

On the complexity of heterogeneous multidimensional quantitative games^{*}

Véronique Bruyère¹, Quentin Hautem^{1**}, and Jean-François Raskin²

¹Université de Mons ²Université Libre de Bruxelles

Abstract. In this paper, we study two-player zero-sum turn-based games played on a finite multi-dimensional weighted graph. In recent papers all dimensions use the same measure, whereas here we allow to combine different measures. Such heterogeneous multidimensional quantitative games provide a general and natural model for the study of reactive system synthesis. We focus on classical measures like the Inf, Sup, LimInf, and LimSup of the weights seen along the play, as well as on the window mean-payoff (WMP) measure recently introduced in [6]. This new measure is a natural strengthening of the mean-payoff measure. We allow objectives defined as Boolean combinations of heterogeneous constraints. While multidimensional games with Boolean combinations of mean-payoff constraints are undecidable [22], we show that the problem becomes EXPTIME-complete for DNF/CNF Boolean combinations of heterogeneous measures taken among {WMP, Inf, Sup, LimInf, LimSup} and that exponential memory strategies are sufficient for both players to win. We provide a detailed study of the complexity and the memory requirements when the Boolean combination of the measures is replaced by an intersection. EXPTIME-completeness and exponential memory strategies still hold for the intersection of measures in {WMP, Inf, Sup, LimInf, LimSup}, and we get PSPACE-completeness when WMP measure is no longer considered. To avoid EXPTIME- or PSPACE-hardness, we impose at most one occurrence of WMP measure and fix the number of Sup measures, and we propose several refinements (on the number of occurrences of the other measures) for which we get polynomial algorithms and lower memory requirements. For all the considered classes of games, we also study parameterized complexity.

1 Introduction

Two-player zero-sum turn-based games played on graphs are an adequate mathematical model to solve the *reactive synthesis problem* [21]. To model systems with resource constraints, like embedded systems, games with quantitative objectives have been studied, e.g. mean-payoff [26] and energy games [3]. In [5,24,6,23], multidimensional games with conjunctions of several quantitative objectives have been investigated, such that all dimensions use the *same* measure. Here, we study games played on multidimensional weighted graphs with objectives defined from different measures over the dimensions. Such *heterogeneous* quantitative games provide a general, natural, and expressive model for the reactive synthesis problem.

We study Boolean combinations of constraints over classical measures like the Inf, Sup, LimInf, and LimSup of the weights seen along the play, as well as the *window mean-payoff* (WMP) measure introduced in [6]. For instance one can ask that along the play the LimInf of the weights on a first component is positive while either the Sup of the weights on a second component or their WMP is less than $\frac{2}{3}$. While the mean-payoff (MP) measure considers the long-run average of the weights along the whole play, the WMP measure considers weights over a *local window of a given size* sliding along the play. A WMP objective asks now to ensure that the average weight satisfies a given constraint over every bounded window. This is a strengthening of the MP objective: winning for the WMP objective implies winning for the MP objective. Also, any finite-memory strategy that forces an MP measure larger than threshold $\nu + \epsilon$ (for any $\epsilon > 0$), also forces the WMP measure to be larger than ν provided that the window size is taken large enough. Aside from their naturalness, WMP objectives are algorithmically more tractable than classical MP objectives, see [6,16]. First, unidimensional WMP games can be solved in polynomial time when working

^{*} This work has been partly supported by ERC Starting Grant (279499: inVEST) and European project Cassting (FP7-ICT-601148).

^{**} Author supported by FRIA fellowship

with polynomial windows [6] while only pseudo-polynomial time algorithms are known for mean-payoff games [26,4]. Second, multidimensional games with Boolean combinations of MP objectives are undecidable [22], whereas we show here that games with Boolean combinations of WMP objectives and other classical objectives are decidable.

Main contributions More precisely (see also Table 1), we show in this paper that the problem is EXPTIME-complete for CNF and DNF Boolean combinations of heterogeneous measures taken among $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. We provide a detailed study of the complexity when the Boolean combination of the measures is replaced by an intersection, as it is often natural in practice to consider conjunction of constraints. EXPTIME-completeness of the problem still holds for the intersection of measures in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$, and we get PSPACE-completeness when WMP measure is not considered. To avoid EXPTIME-hardness, we consider fragments where there is at most one occurrence of a WMP measure. In case of intersections of one WMP objective with any number of objectives of one kind among $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ (this number must be fixed in case of objectives Sup), we get P-completeness when dealing with polynomial windows, a reasonable hypothesis in practical applications. In case of no occurrence of WMP measure, we propose several refinements (on the number of occurrences of the other measures) for which we get P-completeness. Some of our results are obtained by reductions to known qualitative games but most of them are obtained by new algorithms that require genuine ideas to handle in an optimal way one WMP objective together with qualitative objectives such as safety, reachability, Büchi and coBüchi objectives. Finally, in all our results, we provide a careful analysis of the memory requirements of winning strategies for both players, and we study parameterized complexity.

Objectives	Complexity class	Player 1 memory	Player 2 memory
(CNF/DNF) Boolean combination of $\overline{\text{MP}}, \overline{\text{MP}}$ [22]	Undecidable	infinite	infinite
(CNF/DNF) Boolean combinaison of $\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup} (*)$	EXPTIME-complete	exponential	
Intersection of $\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup} (*)$			
Intersection of WMP [6]			
Intersection of $\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup} (*)$ and refinements $(*)$	PSPACE-complete	See Table 4	
Intersection of $\overline{\text{MP}}$ [23]	coNP-complete		
Intersection of $\overline{\text{MP}}$ [23]	$\text{NP} \cap \text{coNP}$	infinite	memoryless
Unidimensional $\overline{\text{MP}}$ [26,4]		memoryless	
Unidimensional WMP [6]	P-complete	pseudo-polynomial	
$\text{WMP} \cap \Omega$ with $\Omega \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\} (*)$	(Polynomial windows)		
Unidimensional $\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}$ [15]	P-complete	memoryless	

Table 1. Overview - Our results are marked with (*)

Related work Multidimensional mean-payoff games have been studied in [23]. Conjunction of lim inf mean-payoff ($\underline{\text{MP}}$) objectives are coNP-complete, conjunctions of lim sup MP ($\overline{\text{MP}}$) objectives are in NP and in coNP, and infinite-memory strategies are necessary for player 1 whereas player 2 can play memoryless. The general case of Boolean combinations of $\underline{\text{MP}}$ and $\overline{\text{MP}}$ have been shown undecidable in [22]. Multidimensional energy games were studied in [5,9] for unfixed initial credit and in [18] for fixed initial credit. For unfixed initial credit, they are coNP-complete and exponential memory strategies are necessary and sufficient to play optimally. For fixed initial credit, they are 2EXPTIME-complete and doubly exponential memory is necessary and sufficient to play optimally. Energy games and mean-payoff games with imperfect information, that generalize multidimensional energy and mean-payoff games, have been studied in [11] and shown undecidable.

The WMP measure was first introduced in [6]. Unidimensional WMP games can be solved in polynomial time, and multidimensional WMP games are EXPTIME-complete. In [6], the WMP measure is considered on all the dimensions, with no conjunction with other measures like Inf, Sup, LimInf, and LimSup, and the case of Boolean combinations of WMP objectives is not investigated.

Games with objectives expressed in fragments of LTL have been studied in [1]. Our result that games with intersection of objectives in $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ are in PSPACE can be obtained by reduction to some of these fragments. But we here propose a simple proof adapted to our context, that allows to identify several polynomial fragments. Our other results cannot be obtained in this way and require genuine techniques and new algorithmic ideas.

Structure of the paper In Section 2, we fix definitions and notations, and recall known results. In Section 3, we study the complexity of multidimensional games with an intersection of objectives taken among $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. In Section 4, we identify a polynomial fragment when only one WMP objective is allowed in the intersection. In Section 5, we study intersections that do not use WMP objectives, and we identify some additional polynomial fragments. In Section 6, we establish the complexity of general Boolean combinations (instead of intersections) of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. In Section 7, we study the parameterized complexity of our results of Sections 3, 5 and 6. Finally, in Section 8, we give a conclusion.

2 Preliminaries

We consider turn-based games played by two players on a finite multidimensional weighted directed graph.

Definition 1. A multi-weighted two-player game structure (or simply game structure) is a tuple $G = (V_1, V_2, E, w)$ where

- (V, E) is a finite directed graph, with V the set of vertices and $E \subseteq V \times V$ the set of edges such that for each $v \in V$, there exists $(v, v') \in E$ for some $v' \in V$ (no deadlock),
- (V_1, V_2) forms a partition of V such that V_p is the set of vertices controlled by player $p \in \{1, 2\}$,
- $w : E \rightarrow \mathbb{Z}^n$ is the n -dimensional weight function that associates a vector of n weights to each edge, for some $n \geq 1$.

We also say that G is an n -weighted game structure. The opponent of player $p \in \{1, 2\}$ is denoted by \bar{p} . A play of G is an infinite sequence $\rho = \rho_0 \rho_1 \dots \in V^\omega$ such that $(\rho_k, \rho_{k+1}) \in E$ for all $k \in \mathbb{N}$. We denote by $\text{Plays}(G)$ the set of plays in G . Histories of G are finite sequences $\rho = \rho_0 \dots \rho_i \in V^+$ defined in the same way. The set of all histories is denoted by $\text{Hist}(G)$. Given a play $\rho = \rho_0 \rho_1 \dots$, the history $\rho_k \dots \rho_{k+i}$ is also denoted by $\rho_{[k, k+i]}$. We denote by w_m the projection of the weight function w on the m^{th} dimension, and by W the maximum weight in absolute value on all dimensions.

Strategies A strategy σ for player $p \in \{1, 2\}$ is a function $\sigma : V^*V_p \rightarrow V$ assigning to each history $hv \in V^*V_p$ a vertex $v' = \sigma(hv)$ such that $(v, v') \in E$. It is *memoryless* if $\sigma(hv) = \sigma(h'v)$ for all histories $hv, h'v$ ending with the same vertex v , that is, σ is a function $\sigma : V_p \rightarrow V$. It is *finite-memory* if it can be encoded by a deterministic *Moore machine* $(M, m_0, \alpha_u, \alpha_n)$ where M is a finite set of states (the memory of the strategy), $m_0 \in M$ is the initial memory state, $\alpha_u : M \times V \rightarrow M$ is an update function, and $\alpha_n : M \times V_p \rightarrow V$ is the next-action function. The Moore machine defines a strategy σ such that $\sigma(hv) = \alpha_n(\hat{\alpha}_u(m_0, h), v)$ for all histories $hv \in V^*V_p$, where $\hat{\alpha}_u$ extends α_u to sequences of vertices as expected. Note that such a strategy is memoryless if $|M| = 1$.

Given a strategy σ of player $p \in \{1, 2\}$, we say that a play ρ of G is *consistent* with σ if $\rho_{k+1} = \sigma(\rho_0 \dots \rho_k)$ for all $k \in \mathbb{N}$ such that $\rho_k \in V_p$. A history consistent with a strategy is defined similarly. Given an initial vertex v_0 , and a strategy σ_p of each player p , we have a unique play that is consistent with both strategies. This play is called the *outcome* of (σ_1, σ_2) from v_0 and is denoted by $\text{Out}(v_0, \sigma_1, \sigma_2)$.

Closed sets and attractors Let $G = (V_1, V_2, E, w)$ be a game structure, and let p be a player in $\{1, 2\}$. A set $U \subseteq V$ is *p-closed* if for all $v \in U \cap V_p$ and all $(v, v') \in E$, we have $v' \in U$ (player p cannot leave U), and for all $v \in U \cap V_{\bar{p}}$, there exists $(v, v') \in E$ such that $v' \in U$ (player \bar{p} can ensure to stay in U). We thus get a subgame structure induced by U that is denoted by $G[U]$.

Given $U \subseteq V$, the p -attractor $\text{Attr}_p(U, G)$ is the set of vertices from which player p has a strategy to reach U against any strategy of player \bar{p} . It is constructed by induction as follows: $\text{Attr}_p(U, G) = \cup_{k \geq 0} X_k$ such that

$$\begin{aligned} X_0 &= U, \\ X_{k+1} &= X_k \cup \{v \in V_p \mid \exists (v, v') \in E, v' \in X_k\} \cup \{v \in V_{\bar{p}} \mid \forall (v, v') \in E, v' \in X_k\}. \end{aligned}$$

Notice that $U \subseteq \text{Attr}_p(U, G)$. The next properties are classical.

Theorem 1. *Let $G = (V_1, V_2, E, w)$ be a game structure, let p be a player in $\{1, 2\}$, and let $U \subseteq V$ be a set of vertices. Then*

- The attractor $\text{Attr}_p(U, G)$ can be computed in $O(|V| + |E|)$ time [2,17].
- The set $V \setminus \text{Attr}_p(U, G)$ is p -closed [25].
- If U is \bar{p} -closed, then $\text{Attr}_p(U, G)$ is \bar{p} -closed [25].

Objectives and winning sets Let $G = (V_1, V_2, E, w)$ be a game structure. An *objective* for player p is a set of plays $\Omega \subseteq \text{Plays}(G)$. An objective can be *qualitative* (it only depends on the graph (V, E)), or *quantitative* (it also depends on the weight function w). A play ρ is *winning* for player p if $\rho \in \Omega$, and losing otherwise (i.e. winning for player \bar{p}). We thus consider zero-sum games such that the objective of player \bar{p} is $\bar{\Omega} = \text{Plays}(G) \setminus \Omega$, that is, opposite to the objective Ω of player p . In the following, we always take the point of view of player 1 by supposing that Ω is his objective, and we denote by (G, Ω) the corresponding game. Given an initial state v_0 , a strategy σ_p for player p is *winning* from v_0 if $\text{Out}(v_0, \sigma_p, \sigma_{\bar{p}}) \in \Omega$ for all strategies $\sigma_{\bar{p}}$ of player \bar{p} . Vertex v_0 is also called *winning* for player p and the *winning set* Win_p^Ω is the set of all his winning vertices. Similarly the winning vertices of player \bar{p} are those from which player \bar{p} can ensure to satisfy his objective $\bar{\Omega}$ against all strategies of player p , and $\text{Win}_{\bar{p}}^{\bar{\Omega}}$ is his winning set. When $\text{Win}_p^\Omega \cup \text{Win}_{\bar{p}}^{\bar{\Omega}} = V$, we say that the game is *determined*. It is known that every turn-based game with Borel objectives is determined [19]. When we need to refer to a specific game structure G , we use notation $\text{Win}_p^\Omega(G)$.

Qualitative objectives Let $G = (V_1, V_2, E)$ be an *unweighted* game structure. Given a set $U \subseteq V$, classical qualitative objectives are the following ones.

- A *reachability* objective $\text{Reach}(U)$ asks to visit a vertex of U at least once.
- A *safety* objective $\text{Safe}(U)$ asks to visit no vertex of $V \setminus U$, that is, to avoid $V \setminus U$. A safety objective is thus the opposite of a reachability objective, that is $\text{Safe}(U) = \overline{\text{Reach}(V \setminus U)}$.
- A *Büchi* objective $\text{Buchi}(U)$ asks to visit a vertex of U infinitely often.
- A *co-Büchi* objective $\text{CoBuchi}(U)$ asks to visit no vertex of $V \setminus U$ infinitely often. A co-Büchi objective is thus the opposite of a Büchi objective, that is $\text{CoBuchi}(U) = \overline{\text{Buchi}(V \setminus U)}$.

These objectives can be mixed by taking their intersection. Let U_1, \dots, U_i be a family of subsets of V :

- A *generalized reachability* objective $\text{GenReach}(U_1, \dots, U_i) = \cap_{k=1}^i \text{Reach}(U_k)$ asks to visit a vertex of U_k at least once, for each $k \in \{1, \dots, i\}$.
- A *generalized Büchi* objective $\text{GenBuchi}(U_1, \dots, U_i) = \cap_{k=1}^i \text{Buchi}(U_k)$ asks to visit a vertex of U_k infinitely often, for each $k \in \{1, \dots, i\}$.
- A *generalized Büchi \cap co-Büchi* objective $\text{GenBuchi}(U_1, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i)$ asks to visit a vertex of U_k infinitely often, for each $k \in \{1, \dots, i-1\}$, and no vertex of $V \setminus U_i$ infinitely often.

Notice that the intersection of safety objectives $\cap_{k=1}^i \text{Safe}(U_k)$ is again a safety objective that is equal to $\text{Safe}(\cap_{k=1}^i U_k)$. Similarly the intersection of co-Büchi objectives $\cap_{k=1}^i \text{CoBuchi}(U_k)$ is the co-Büchi objective $\text{CoBuchi}(\cap_{k=1}^i U_k)$. Notice also that in an intersection of objectives, a safety objective $\text{Safe}(U)$ can be replaced by the co-Büchi objective $\text{CoBuchi}(U)$ by making each vertex of $V \setminus U$ absorbing, that is, with a unique outgoing edge being a self loop.

A game with an objective Ω is just called Ω game. Hence a game with a reachability objective is called *reachability game*, aso. In the sequel, we sometimes say by abuse of notation that $\Omega \in \{\text{Reach}, \text{Safe}, \text{Buchi}, \text{CoBuchi}\}$ without mentioning a subset U of vertices.

As all the previous qualitative objectives Ω are ω -regular and thus Borel objectives, the corresponding games (G, Ω) are determined.

Quantitative objectives For a 1-weighted game structure $G = (V_1, V_2, E, w)$ (with dimension $n = 1$), let us now introduce quantitative objectives defined by some classical *measure functions* $f : \text{Plays}(G) \rightarrow \mathbb{Q}$. Such a function f associates a rational number to each play $\rho = \rho_1 \rho_2 \dots$ according to the weights $w(\rho_k, \rho_{k+1})$, $k \geq 0$, and can be one among the next functions. Let $\rho \in \text{Plays}(G)$:

- $\text{Inf}(\rho) = \inf_{k \geq 0} (w(\rho_k, \rho_{k+1}))$: the Inf measure defines the minimum weight seen along the play.
- $\text{Sup}(\rho) = \sup_{k \geq 0} (w(\rho_k, \rho_{k+1}))$: the Sup measure defines the maximum weight seen along the play.
- $\text{LimInf}(\rho) = \liminf_{k \rightarrow \infty} (w(\rho_k, \rho_{k+1}))$: the LimInf measure defines the minimum weight seen infinitely often along the play.
- $\text{LimSup}(\rho) = \limsup_{k \rightarrow \infty} (w(\rho_k, \rho_{k+1}))$: the LimSup measure defines the maximum weight seen infinitely often along the play.

Given such a measure function $f \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$, a bound $\nu \in \mathbb{Q}$, and a relation $\sim \in \{>, \geq, <, \leq\}$, we define the objective $\Omega = f(\sim \nu)$ such that

$$f(\sim \nu) = \{\rho \in \text{Plays}(G) \mid f(\rho) \sim \nu\}. \quad (1)$$

We are also interested in the next two measure functions defined on histories instead of plays. Let $\rho = \rho_0 \dots \rho_i \in \text{Hist}(G)$:

- $\text{TP}(\rho) = \sum_{k=0}^{i-1} w(\rho_k, \rho_{k+1})$: the *total-payoff* measure TP defines the sum of the weights seen along the history.
- $\text{MP}(\rho) = \frac{1}{i} \text{TP}(\rho)$: the *mean-payoff* measure MP defines the mean of the weights seen along the history.

The second measure can be extended to plays ρ as either $\underline{\text{MP}}(\rho) = \liminf_{k \geq 0} \text{MP}(\rho_{[0,k]})$ or $\overline{\text{MP}}(\rho) = \limsup_{k \geq 0} \text{MP}(\rho_{[0,k]})$. The MP measure on histories allows to define the *window mean-payoff objective*, a new ω -regular objective introduced in [6]: given a bound $\nu \in \mathbb{Q}$, a relation $\sim \in \{>, \geq, <, \leq\}$, and a window size $\lambda \in \mathbb{N} \setminus \{0\}$, the objective $\text{WMP}(\lambda, \sim \nu)$ ¹ is equal to

$$\text{WMP}(\lambda, \sim \nu) = \{\rho \in \text{Plays}(G) \mid \forall k \geq 0, \exists l \in \{1, \dots, \lambda\}, \text{MP}(\rho_{[k, k+l]}) \sim \nu\}. \quad (2)$$

The window mean-payoff objective asks that the average weight becomes $\sim \nu$ inside a local bounded window for all positions of this window sliding along the play, instead of the classical mean-payoff objective asking that the long run-average $\underline{\text{MP}}(\rho)$ (resp. $\overline{\text{MP}}(\rho)$) is $\sim \nu$. This objective is a strengthening of the mean-payoff objective.

Given a multi-weighted game structure $G = (V_1, V_2, E, w)$ (with dimension $n \geq 1$), we can mix objectives of (1) and (2) by fixing one such objective Ω_m for each dimension $m \in \{1, \dots, n\}$, and then taking the intersection $\cap_{m=1}^n \Omega_m$. More precisely, given a vector $(\sim_1 \nu_1, \dots, \sim_n \nu_n)$, each objective Ω_m uses a measure function based on the weight function w_m ; Ω_m is either of the form $f(\sim_m \nu_m)$ with $f \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$, or of the form $\text{WMP}(\lambda, \sim_m \nu_m)$ for some window size λ (this size can change with m).

As done for qualitative objectives, we use the shortcut Ω game for a game with quantitative objective Ω . For instance an $\text{Inf}(\sim \nu)$ game is a 1-weighted game with objective $\text{Inf}(\sim \nu)$, a $\text{LimSup}(\sim_1 \nu_1) \cap \text{WMP}(\lambda, \sim_2 \nu_2) \cap \text{Inf}(\sim_3 \nu_3)$ game is a 3-weighted game with the intersection of a $\text{LimSup}(\sim_1 \nu_1)$ objective on the first dimension, a $\text{WMP}(\lambda, \sim_2 \nu_2)$ objective on the second dimension, and an $\text{Inf}(\sim_3 \nu_3)$ objective on the third dimension.

In the sequel, we sometimes abusively say that $\Omega = \cap_{m=1}^n \Omega_m$ with $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ without mentioning the used relations, bounds and window sizes. It is implicitly supposed that Ω_m deals with the m^{th} component of the weight function.

In this paper, we want to study the next problem. We will go beyond this problem in Section 6 by considering Boolean combinations (instead of conjunctions) of objectives.

¹ This objective is called “direct fixed window mean-payoff” in [6] among several other variants.

Problem 1. Let $G = (V_1, V_2, E, w)$ be a multi-weighted game structure with dimension $n \geq 1$, and $\Omega = \bigcap_{m=1}^n \Omega_m$ be a quantitative objective such that each $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. Can we compute the winning sets Win_1^Ω and Win_2^Ω ? If yes what is the complexity of computing these sets and how (memoryless, finite-memory, general) are the winning strategies of both players? Given such a game (G, Ω) and an initial vertex v_0 , the *synthesis problem* asks to decide whether player 1 has a winning strategy for Ω from v_0 and to build such a strategy when it exists.

Remark 1. (i) In this problem, we can assume that the bounds used in the vector $(\sim_1 \nu_1, \dots, \sim_n \nu_n)$ are such that $(\nu_1, \dots, \nu_n) = (0, \dots, 0)$. Indeed, suppose that $\nu_m = \frac{a}{b}$ with $a \in \mathbb{Z}$ and $b \in \mathbb{N} \setminus \{0\}$, then replace the m^{th} component of the weight function w by $b \cdot w_m - a$. (ii) Moreover notice that if $\nu_m = 0$ and $\Omega_m = \text{WMP}$, then $\text{MP}(\rho_{[k, k+l]})$ can be replaced by $\text{TP}(\rho_{[k, k+l]})$ in (2). (iii) Finally, the vector $(\sim_1 0, \dots, \sim_n 0)$ can be supposed to be equal to $(\geq 0, \dots, \geq 0)$. Indeed strict inequality > 0 (resp. < 0) can be replaced by inequality ≥ 1 (resp. ≤ -1), and inequality ≤ 0 can be replaced by ≥ 0 by replacing the weight function by its negation and the measure Inf (resp. Sup , LimInf , LimSup , TP) by Sup (resp. Inf , LimSup , LimInf , TP).

From now on, we only work with vectors $(\geq 0, \dots, \geq 0)$ and we no longer mention symbol \geq . Hence, as an example, $\text{Inf}(\geq 0)$ and $\text{WMP}(2, \geq 0)$ are replaced by $\text{Inf}(0)$ and $\text{WMP}(2, 0)$; and when the context is clear, we only mention Inf and WMP .

All the objectives considered in Problem 1 are ω -regular, the corresponding games (G, Ω) are thus determined. In the proofs of this paper, we will often use the property that $\text{Win}_p^\Omega \cup \text{Win}_p^{\overline{\Omega}} = V$.

Let us illustrate a mixing of quantitative objectives on the following example.

Example 1. Consider the 3-weighted game structure depicted on Figure 1. In all examples in this paper, we assume that circle (resp. square) vertices are controlled by player 1 (resp. player 2). Let $\Omega = \text{WMP}(3, 0) \cap \text{Sup}(0) \cap \text{LimSup}(0)$ be the objective of player 1. Let us recall that by definition of Ω , we look at the WMP (resp. Sup , LimSup) objective on the first (resp. second, third) dimension. Let us show that v_0 is a winning vertex for player 1. Let σ_1 be the following strategy of player 1 from v_0 : go to v_1 , take the self loop once, go back to v_0 and then always go to v_2 . More precisely, we have $\sigma_1(v_0) = v_1$, $\sigma_1(v_0 v_1) = v_1$, $\sigma_1(v_0 v_1 v_1) = v_0$, and $\sigma_1(v_0 v_1 v_1 h v_0) = v_2$ for all $h v_0 \in V^* V_1$. Let σ_2 be any strategy of player 2. We are going to prove that $\rho = \text{Out}(v_0, \sigma_1, \sigma_2) \in \Omega$. Notice that $\rho \in v_0 v_1 v_1 v_0 \{v_2, v_0\}^\omega$. As player 1 forces ρ to begin with $v_0 v_1 v_1$, he ensures that $\text{Sup}(\rho) \geq 0$ on the second component. Moreover as ρ visits infinitely often edge (v_2, v_2) or (v_2, v_0) , player 1 also ensures to have $\text{LimSup}(\rho) \geq 0$ on the third component. Finally, we have to check that $\rho \in \text{WMP}(3, 0)$ with respect to the first component, that is (by Remark 1), for all k , there exists $l \in \{1, 2, 3\}$ such that $\text{TP}(\rho_{[k, k+l]}) \geq 0$. For $k = 0$ (resp. $k = 1, k = 2, k = 3$) and $l = 3$ (resp. $l = 2, l = 1, l = 1$), we have $\text{TP}(\rho_{[k, k+l]}) \geq 0$. Now, from position $k = 4$, the sum of weights is non-negative in at most 2 steps. Indeed, either player 2 takes the self loop (v_2, v_2) or he goes to v_0 where player 1 goes back to v_2 . Therefore, for each $k \geq 4$, either $\rho_k = v_0$ and $\text{TP}(\rho_{[k, k+l]}) \geq 0$ with $l = 1$, or $\rho_k = v_2$ and $\text{TP}(\rho_{[k, k+l]}) \geq 0$ with $l = 1$ if $\rho_{k+1} = v_2$, and with $l = 2$ otherwise. It follows that $v_0 \in \text{Win}_1^\Omega$. The strategy σ_1 needs memory: indeed, player 1 needs to remember if he has already visited the edge (v_1, v_1) as this is the only edge visiting a weight greater than or equal to 0 for the Sup objective. Finally, one can show that $\text{Win}_1^\Omega = \{v_0, v_1\}$ and $\text{Win}_2^\Omega = \{v_2\}$.

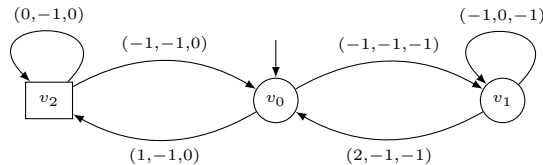


Fig. 1. Example of a multi-weighted two-player game

Remark 2. In Problem 1, the vector $(\sim_1 \nu_1, \dots, \sim_n \nu_n)$ can be assumed equal to $(\geq 0, \dots, \geq 0)$ by Remark 1. It follows that an Inf (resp. Sup, LimInf, LimSup) objective is nothing else than a safety (resp. reachability, co-Büchi, Büchi) objective, and conversely. These two game reductions, together with a third reduction, will be described and proved in Section 3. They all will be used throughout this paper.

Some well-known properties Let us recall some well-known properties about the objectives mentioned above. The complexity of the algorithms is expressed in terms of the size $|V|$ and $|E|$ of the game structure G , the maximum weight W (in absolute value) and the dimension n of the weight function when G is weighted, the number i of objectives in an intersection of objectives², and the window size λ . We begin with the qualitative objectives in an unweighted game structure (see also Table 2).

Theorem 2. – *For reachability or safety games, deciding the winner is P-complete (with an algorithm in $O(|V| + |E|)$ time) and both players have memoryless winning strategies [2,15,17].*
– *For Büchi or co-Büchi games, deciding the winner is P-complete (with an algorithm in $O(|V|^2)$ time) and both players have memoryless winning strategies [7,13,17].*
– *For generalized reachability games, deciding the winner is PSPACE-complete (with an algorithm in $O(2^i \cdot (|V| + |E|))$ time) and exponential memory strategies are necessary and sufficient for both players [14].*
– *For generalized Büchi games, deciding the winner is P-complete (with an algorithm in $O(i \cdot |V| \cdot |E|)$ time) and polynomial memory (resp. memoryless) strategies are necessary and sufficient for player 1 (resp. player 2) [12].*
– *For Büchi \cap co-Büchi games, deciding the winner is P-complete (with an algorithm in $O(|V| \cdot |E|)$ time) and memoryless strategies are necessary and sufficient for both players [10].*
– *For generalized Büchi \cap co-Büchi games, deciding the winner is P-complete (with an algorithm in tile $O(i^2 \cdot |V| \cdot |E|)$) and polynomial memory (resp. memoryless) strategies are necessary and sufficient for player 1 (resp. player 2).³*

In case of reachability games with objective $\text{Reach}(U)$ for player 1, notice that his winning set is equal to $\text{Attr}_1(U, G)$ (see Theorem 1).

Objective	Complexity class	Algorithmic complexity	Player 1 memory	Player 2 memory
Reach/Safe	P-complete	$O(V + E)$	memoryless	memoryless
Büchi/CoBüchi	P-complete	$O(V ^2)$	memoryless	memoryless
GenReach	PSPACE-complete	$O(2^i \cdot (V + E))$	exponential memory	exponential memory
GenBüchi	P-complete	$O(i \cdot V \cdot E)$	polynomial memory	memoryless
Büchi \cap CoBüchi	P-complete	$O(V \cdot E)$	memoryless	memoryless
GenBüchi \cap CoBüchi	P-complete	$O(i^2 \cdot V \cdot E)$	polynomial memory	memoryless

Table 2. Overview of some known results for qualitative objectives (i is the number of objectives in the intersection of reachability/Büchi objectives)

The next corollary will be useful in the proofs.

Corollary 1. [15]

- *For reachability or generalized reachability games, $\text{Win}_2^{\overline{\Omega}}$ is 1-closed. If the set(s) to reach is (are) 2-closed, then Win_1^{Ω} is 2-closed.*
- *For safety games, Win_1^{Ω} is 2-closed. If the set to avoid is 1-closed, then $\text{Win}_2^{\overline{\Omega}}$ is 1-closed.*
- *For Büchi, co-Büchi, generalized Büchi games, and generalized Büchi \cap co-Büchi games, Win_1^{Ω} is 2-closed and $\text{Win}_2^{\overline{\Omega}}$ is 1-closed.*

² Notice that $i = n$ for the objectives considered in Problem 1.

³ This result is obtained thanks to a classical reduction of generalized Büchi \cap co-Büchi games to Büchi \cap co-Büchi games (see also Lemma 6).

We conclude this section with the known results about the window mean-payoff objective.

Theorem 3. [6]⁴ *Let (G, Ω) be an n -weighted game such that $\Omega = \cap_{m=1}^n \Omega_m$ with $\Omega_m = \text{WMP}$ for all m . Then the synthesis problem is EXPTIME-complete (with an algorithm in $O(\lambda^{4n} \cdot |V|^2 \cdot W^{2n})$ time), exponential memory strategies are sufficient and necessary for both players. This problem is already EXPTIME-hard when $n = 2$.*

When $n = 1$, the synthesis problem is decidable in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, both players require finite-memory strategies, and memory in $O(\lambda^2 \cdot W)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|)$) is sufficient for player 1 (resp. player 2). Moreover, if λ is polynomial in the size of the game, then the synthesis problem is P-complete.

3 Intersection of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$

In this section, we solve Problem 1: the synthesis problem is EXPTIME-complete, and exponential memory strategies are both sufficient and necessary for both players (Theorem 4). Before proving this theorem, we first need to introduce some terminology about the window mean-payoff objective. We also need to establish several reductions between different classes of games. These preliminaries will be useful in the proof of various results of this paper, included Theorem 4.

Theorem 4. *Let (G, Ω) be an n -weighted game such that $\Omega = \cap_{m=1}^n \Omega_m$ with $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all m . Then, the synthesis problem is EXPTIME-complete (with an algorithm in $O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$ time), and exponential memory strategies are both necessary and sufficient for both players.*

3.1 Properties of windows

Let us come back to the $\text{WMP}(\lambda, 0)$ objective (with TP in (2)) and let us introduce some useful terminology. Let $\rho = \rho_0 \rho_1 \dots$ be a play. A λ -window at position k is a window of size λ placed along ρ from k to $k + \lambda$. If there exists $l \in \{1, \dots, \lambda\}$ such that $\text{TP}(\rho_{[k, k+l]}) \geq 0$, such a λ -window at position k is called *good* or *closed in $k + l$* (to specify index l), otherwise it is called *bad*. Moreover if l is the smallest index such that $\text{TP}(\rho_{[k, k+l]}) \geq 0$, we say it is *first-closed in $k + l$* .

An interesting property is the following one: a λ -window at position k is *inductively-closed in $k + l$* if it is closed in $k + l$ and for all $k' \in \{k + 1, \dots, k + l - 1\}$, the λ -window at position k' is also closed in $k + l$ (see Figure 2). One easily checks the next property:

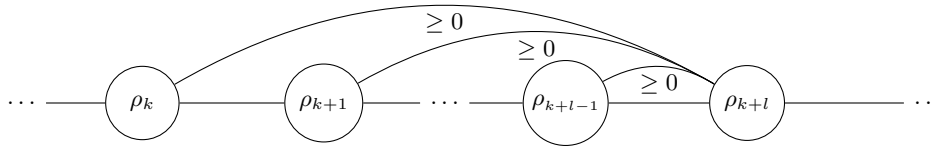


Fig. 2. A λ -window at position k that is inductively-closed in $k + l$

Lemma 1. *If a λ -window at position k is first-closed in $k + l$, then it is inductively-closed in $k + l$. Conversely, if a λ -window at position k is inductively-closed in $k + l$, then either it is first-closed in $k + l$, or there exists $l' \in \{1, \dots, l - 1\}$ such that the λ -window at position k is first-closed in $k + l'$ and the λ -window at position $k + l'$ is inductively-closed in $k + l$.*

The next lemma will be useful in several proofs of this paper:

⁴ When $n = 1$, the time complexity and the memory requirements have been here correctly stated.

Lemma 2. A play ρ is winning for $\text{WMP}(\lambda, 0)$ iff there exists a sequence $(k_i)_{i \geq 0}$ with $k_0 = 0$ such that for each i , we have $k_{i+1} - k_i \in \{1, \dots, \lambda\}$, and the λ -window at position k_i is inductively-closed in k_{i+1} .

When such a sequence $(k_i)_{i \geq 0}$ exists for a play ρ , we say that it is a λ -good decomposition of ρ . We extend this notion to histories $\rho = \rho_0 \rho_1 \dots \rho_k$ as follows. A finite sequence $(k_i)_{i=0}^j$ is a λ -good decomposition of ρ if $k_0 = 0$, $k_j = k$, and for each $i \in \{0, \dots, j-1\}$, we have $k_{i+1} - k_i \in \{1, \dots, \lambda\}$, and the λ -window at position k_i is inductively-closed in k_{i+1} . The size of the decomposition is j . In particular the history ρ_0 (take $k = 0$) has always a λ -good decomposition of size $j = 0$.

Notice that if a sequence $(k_i)_{i \geq 0}$ is a λ -good decomposition of a play ρ , then by Lemma 1, there exists a unique sequence $(k'_i)_{i \geq 0}$ of which $(k_i)_{i \geq 0}$ is a subsequence that is also a λ -good decomposition of ρ and such that for each i the λ -window at position k'_i is first-closed in k'_{i+1} (instead of being only inductively-closed in k'_{i+1}). See also Figure 3. We have a similar property for finite good decompositions. We call such a sequence a *maximal* λ -good decomposition.

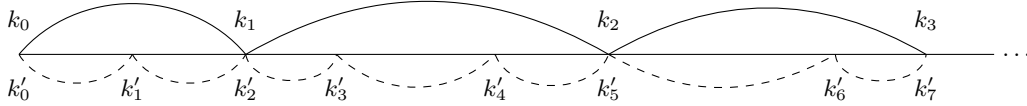


Fig. 3. The sequence $(k'_i)_{i \geq 0}$ is a good decomposition with all its λ -windows being first-closed.

Proof (of Lemma 2). Suppose that ρ is winning for $\text{WMP}(\lambda, 0)$. Then there exists k_1 such that λ -window at position $k_0 = 0$ is first-closed in k_1 (and thus inductively-closed in k_1), there exists k_2 such that λ -window at position k_1 is first-closed in k_2 (and thus first-closed in k_2), aso. This leads to a λ -good decomposition $(k_i)_{i \geq 0}$ of ρ , that is its maximal λ -good decomposition.

Conversely, if there exists a λ -good decomposition $(k_i)_{i \geq 0}$ of ρ , then for each k_i , the λ -window at position k_i is good as well as all λ -windows at position $k' \in \{k_i + 1, \dots, k_{i+1} - 1\}$. It follows that ρ is winning for $\text{WMP}(\lambda, 0)$. \square

Let us come back to Example 1 to illustrate those properties of windows.

Example 1 (continued). We consider again the 3-weighted game structure depicted on Figure 1 and the objective $\Omega = \text{WMP}(3, 0) \cap \text{Sup}(0) \cap \text{LimSup}(0)$ for player 1. We keep the same strategy σ_1 for player 1 from v_0 . Against any strategy of player 2, we have an outcome $\rho \in v_0 v_1 v_1 v_0 \{v_2, v_0\}^\omega$. Let us explain again that $\rho \in \text{WMP}(3, 0)$ with respect to the first component. To this end, we use Lemma 2 and show that ρ has a 3-good decomposition $(k_i)_{i \geq 0}$ (the sequence that we will propose is the maximal 3-good decomposition of ρ). We let $k_0 = 0$, $k_1 = 3$ as the 3-window at position 0 is first-closed in 3, $k_2 = 4$ as the 3-window at position 3 is first-closed in 4. Now, from position 4 the sum of weights is non-negative in at most 2 steps. Indeed, either player 2 takes the self loop (v_2, v_2) or he goes to v_0 where player 1 goes back to v_2 . Then, for $j \geq 3$, we take $k_j = k_{j-1} + 1$ or $k_j = k_{j-1} + 2$, depending on the choice of player 2. This shows that $\rho \in \text{WMP}(3, 0)$ by Lemma 2.

From now on, a λ -window will be simply called a window.

3.2 Game reductions

Let us turn to the reductions. We begin by a polynomial reduction of the class of games $(G, \Omega = \cap_{m=1}^n \Omega_m)$ with each $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ to the class of games $(G', \Omega' = \cap_{m=1}^n \Omega'_m)$ with each $\Omega'_m \in \{\text{WMP}, \text{Safe}, \text{Reach}, \text{CoBuchi}, \text{Buchi}\}$. The objective Ω' is thus the intersection of qualitative and quantitative objectives.

Proposition 1. *Each n -weighted game (G, Ω) with $G = (V_1, V_2, E, w)$, and $\Omega = \cap_{m=1}^n \Omega_m$ such that for all m ,*

$$\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$$

can be polynomially reduced to a game (G', Ω') with $|V| + |E|$ vertices and $2 \cdot |E|$ edges, and $\Omega' = \cap_{m=1}^n \Omega'_m$ such that for all m ,

$$\Omega'_m \in \{\text{WMP}, \text{Safe}, \text{Reach}, \text{CoBuchi}, \text{Buchi}\}$$

Moreover, if Ω_m is a WMP (resp. Inf, Sup, LimInf, LimSup) objective, then Ω'_m is a WMP (resp. Safe, Reach, CoBuchi, Buchi) objective.

A memoryless (resp. (polynomial, exponential) memory) strategy in G' transfers to a memoryless (resp. (polynomial, exponential) memory) strategy in G .

Proof. Let (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ such that $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all m . By Remark 1, we suppose that the threshold ν is equal $(0, \dots, 0)$ and that $\text{MP}(\rho_{[k, k+l]})$ is replaced by $\text{TP}(\rho_{[k, k+l]})$ in (2).

From G , we construct a new game structure $G' = (V'_1, V'_2, E')$ in a way to have weights depending on vertices instead of edges, except for objectives $\Omega_m = \text{WMP}$: each edge $e = (v, v') \in E$ is split into two consecutive edges, and the new intermediate vertex belongs to player 1⁵. The resulting game structure has thus $|V| + |E|$ vertices and $2 \cdot |E|$ edges, that is, a polynomial size. To explain how to proceed with the weights, we first suppose that G is 1-weighted and $\Omega = \Omega_1$. Then we will explain how to extend the construction to n -weighted game structures and $\Omega = \cap_{m=1}^n \Omega_m$.

Let $n = 1$. We proceed as follows in the case $\Omega = \text{Inf}(0)$. For each edge $e \in E$ split into two consecutive edges, the new intermediate vertex is decorated with 0 if $w(e) \geq 0$ and with -1 otherwise, and the vertices of V are decorated with 0. Then, we define the set $U' \subseteq V'$ composed of all vertices decorated with 0. Clearly, player 1 has a winning strategy from vertex v_0 in the initial game iff player 1 has a winning strategy from v_0 in the constructed game with the safety objective $\Omega' = \text{Safe}(U')$. If $\Omega = \text{Sup}(0)$ (resp. $\text{LimInf}(0)$, $\text{LimSup}(0)$), the construction is the same except that the vertices of V are decorated with -1 (resp. 0, -1) and the objective Ω' is now $\text{Reach}(U')$ (resp. $\text{CoBuchi}(U')$, $\text{Buchi}(U')$).

When $n = 1$, it remains to consider the case $\Omega = \text{WMP}(\lambda, 0)$ with $\lambda \in \mathbb{N} \setminus \{0\}$. For each edge $e \in E$ split into two consecutive edges, the weight $w(e)$ is replaced by two weights respectively equal to -1 on the first new edge and $1 + w(e)$ on the second one. Let us show that player 1 has a winning strategy from vertex v_0 in the initial game iff player 1 has a winning strategy from v_0 in the constructed game with the $\text{WMP}(2\lambda, 0)$ objective. There is a one-to-one correspondance between the plays of G and the plays of G' . Let $\rho \in \text{Plays}(G)$ and its corresponding $\rho' \in \text{Plays}(G')$. If a window at position k is closed in $k + l$ for ρ , then the window of size 2λ at position $2k$ (resp. $2k + 1$) is closed in $2k + 2l$ for ρ' . It follows that if ρ is winning for the $\text{WMP}(\lambda, 0)$ objective, then ρ' is winning for the $\text{WMP}(2\lambda, 0)$ objective. The converse is also true since if a window at position $2k$ is closed in $2k + l$ for ρ' , then we can suppose that $l = 2l'$, and thus we have a window at position k that is closed in $k + l'$ for ρ .

Let us now suppose that $n \geq 2$. Let n^* be the number of objectives $\Omega_m \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ and $n - n^*$ be the number of objectives $\Omega_m = \text{WMP}$. We proceed as in the case $n = 1$ except that the vertices of the new game structure are decorated by vectors of weights in $\{-1, 0\}^{n^*}$ and its edges are labeled by vectors of weights in \mathbb{Z}^{n-n^*} . The decoration and labeling are done component-wise as explained above. The game structure G' has still a size polynomial in the initial game structure G .

Finally, it is clear that any strategy in G' that is memoryless (resp. (polynomial, exponential) memory) transfers to a strategy in G that is again memoryless (resp. (polynomial, exponential) memory). \square

We now turn to an exponential reduction of the class of games $(G, \Omega = \cap_{m=1}^n \Omega_m)$ with each $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ to the class of games $(G', \Omega' = \cap_{m=1}^n \Omega'_m)$ with each Ω'_m being an objective in $\{\text{Buchi}, \text{CoBuchi}\}$.

Proposition 2. *Each n -weighted game (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ such that for all m ,*

$$\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$$

⁵ It could belong to player 2 since there is exactly one outgoing edge.

can be exponentially reduced to a game (G', Ω') with $O(|V| \cdot (\lambda^2 \cdot W)^n)$ vertices and $O(|E| \cdot (\lambda^2 \cdot W)^n)$ edges, and $\Omega' = \cap_{m=1}^n \Omega'_m$ such that for all m ,

$$\Omega'_m \in \{\text{Buchi}, \text{CoBuchi}\}$$

Moreover, if Ω_m is a WMP (resp. Inf, Sup, LimInf, LimSup) objective, then Ω'_m is a CoBuchi (resp. CoBuchi, Buchi, CoBuchi, Buchi) objective.

A memoryless (resp. finite-memory) strategy in G' transfers to a finite-memory strategy in G .

Proof. Let (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ such that $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all m . We proceed like in the previous proof by first considering 1-weighted games.

Let $n = 1$ and $\Omega = \text{Inf}(0)$. From G , we construct a new unweighted game structure $G' = (V_1', V_2', E')$ in a way to keep in the current vertex of G' the information whether or not we have seen a strictly negative weight along the current history in $\text{Hist}(G)$. More precisely, we define $V' = V \times \{-1, 0\}$ such that -1 (resp. 0) expresses the presence (resp. absence) of strictly negative weights, and E' is composed of the edges $((v, a), (v', a'))$ such that $e = (v, v') \in E$ and $a' = -1$ if $w(e) < 0$ or $a = -1$, and $a' = 0$ otherwise. Then, we define the set $U' \subseteq V'$ composed of all vertices in $V \times \{0\}$. It follows that player 1 has a winning strategy from v_0 in (G, Ω) iff player 1 has a winning strategy from $(v_0, 0)$ in (G', Ω') with $\Omega' = \text{CoBuchi}(U')$.

The case $\Omega = \text{Sup}(0)$ is solved similarly with the current vertex of G' now keeping the information whether or not we have seen a positive weight along the current history in $\text{Hist}(G)$. We define $V' = V \times \{-1, 0\}$ such that 0 (resp. -1) expresses the presence (resp. absence) of positive weights; E' is composed of the edges $((v, a), (v', a'))$ such that $e = (v, v') \in E$ and $a' = 0$ if $w(e) \geq 0$ or $a = 0$, and $a' = -1$ otherwise; and $U' = V \times \{0\}$. Then player 1 has a winning strategy from v_0 in (G, Ω) iff player 1 has a winning strategy from $(v_0, -1)$ in (G', Ω') with $\Omega' = \text{Buchi}(U')$.

The cases $\Omega = \text{LimInf}(0)$ and $\Omega = \text{LimSup}(0)$ are solved more easily. Again $V' = V \times \{-1, 0\}$, but now $((v, a), (v', a')) \in E'$ iff $e = (v, v') \in E$ and $a' = 0$ if $w(e) \geq 0$, $a' = -1$ otherwise (we just remember the sign of the current weight in G). The $\text{LimInf}(0)$ (resp. LimSup) objective is replaced by the $\text{CoBuchi}(U')$ (resp. $\text{Buchi}(U')$) objective with $U' = V \times \{0\}$.

It remains to study the case $\Omega = \text{WMP}(\lambda, 0)$ when $n = 1$. We proceed like in [6]. By Lemma 2, a play ρ is winning iff ρ has a λ -good decomposition. This good decomposition can be supposed maximal by Lemma 1. Hence ρ is winning iff there exists a sequence $(k_i)_{i \geq 0}$ with $k_0 = 0$ such that for each i , we have $k_{i+1} - k_i \in \{1, \dots, \lambda\}$, and the window at position k_i is first-closed in k_{i+1} . We thus keep in the current vertex of G' the sum of the weights of the current history in $\text{Hist}(G)$, and as soon as it turns non-negative (in at most λ steps), we reset the sum counter to zero. The number of performed steps is also stored in the current vertex as a second counter. More precisely, we define $V' = V \times \{-\lambda \cdot W, \dots, 0\} \times \{0, \dots, \lambda - 1\} \cup \{\beta\}$, where the additional vertex β denotes a special vertex for the detection of a bad window (the current sum is negative and the number of steps exceeds λ). Let $u = (v, s, l) \in V'$ be such that $v \in V$, s is the current sum, and l is the current number of steps. Given $e = (v, v') \in E$, we construct the edge $(u, u') \in E'$ outgoing u such that

$$u' = \begin{cases} (v', s + w(e), l + 1) & \text{if } s + w(e) < 0 \text{ and } l < \lambda - 1 \\ \beta & \text{if } s + w(e) < 0 \text{ and } l = \lambda - 1 \\ (v', 0, 0) & \text{if } s + w(e) \geq 0 \end{cases}$$

We also add the edge (β, β) to E' , and define $U' = V' \setminus \{\beta\}$. Clearly, player 1 has a winning strategy from v_0 in (G, Ω) iff player 1 has a winning strategy from $(v_0, 0, 0)$ in (G', Ω') with $\Omega' = \text{CoBuchi}(U')$.

Let us turn to n -weighted games G with $n \geq 2$. Let n^* be the number of objectives $\Omega_m \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ and $n - n^*$ be the number of objectives $\Omega_m = \text{WMP}$. We proceed as in the case $n = 1$ by adding some information in the vertices for each component $m \in \{1, \dots, n\}$:

$$V' = V \times \{-1, 0\}^{n^*} \times \left(\{-\lambda \cdot W, \dots, 0\} \times \{0, \dots, \lambda - 1\} \cup \{\beta\} \right)^{n - n^*}. \quad (3)$$

The edges of E' are defined component-wise as explained above⁶. The game structure G' has a size exponential in the initial game structure G , that is, with $O(|V'|) = O(|V| \cdot (\lambda^2 \cdot W)^n)$ vertices and $O(|E'|) = O(|E| \cdot (\lambda^2 \cdot W)^n)$ edges.

Finally, any strategy in G' that is memoryless or finite-memory transfers to a strategy in G that has to be finite-memory in both cases. \square

The last reduction is concerned with a polynomial reduction of the class of games $(G, \Omega = \cap_{m=1}^n \Omega_m)$ with each $\Omega_m \in \{\text{Reach, Safe, Buchi, CoBuchi}\}$ to the class of games $(G', \Omega' = \cap_{m=1}^n \Omega'_m)$ with each $\Omega'_m \in \{\text{Sup, Inf, LimSup, LimInf}\}$.

Proposition 3. *Each unweighted game (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ such that for all m ,*

$$\Omega_m \in \{\text{Reach, Safe, Buchi, CoBuchi}\}$$

can be polynomially reduced to an n -weighted game (G', Ω') with $|V|$ vertices and $|E|$ edges, and $\Omega' = \cap_{m=1}^n \Omega'_m$, such that for all m ,

$$\Omega'_m \in \{\text{Sup, Inf, LimSup, LimInf}\}$$

Moreover, if Ω_m is a Reach (resp. Safe, Buchi, CoBuchi) objective, then Ω'_m is a Sup (resp. Inf, LimSup, LimInf) objective.

A memoryless (resp. (polynomial, exponential) memory) strategy in G' transfers to a memoryless (resp. (polynomial, exponential) memory) strategy in G .

Proof. Let (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ such that $\Omega_m \in \{\text{Reach, Safe, Buchi, CoBuchi}\}$ for all m . We proceed like in the previous proofs by first considering 1-weighted games.

Let $n = 1$ and $\Omega = \text{Reach}(U)$ (resp. $\text{Safe}(U)$, $\text{Buchi}(U)$, $\text{CoBuchi}(U)$). We construct the new game (G', Ω') as follows. The game structure G' is the game structure G with a weight function w such that, for all $e = (v, v') \in E$, $w(e) = 0$ if $v \in U$, -1 otherwise. We let $\Omega' = \text{Sup}(0)$ (resp. $\text{Inf}(0)$, $\text{LimSup}(0)$, $\text{LimInf}(0)$). Thus, the game G' is built in a way to necessarily see a non-negative weight iff one visits a vertex of U in G . Clearly, player 1 has a winning strategy from v_0 in the (G, Ω) iff he has a winning strategy from v_0 in (G', Ω') .

Let us now suppose that $n \geq 2$. We proceed as in the case $n = 1$ except that the weight function is n -dimensional, and is defined component-wise as explained above. Let us note that the game structure G' has still the same size as G .

Finally, any strategy in G' that is memoryless (resp. (polynomial, exponential) memory) transfers to a strategy in G that is again memoryless (resp. (polynomial, exponential) memory). \square

3.3 Proof of Theorem 4

We here provide the proof of Theorem 4 which uses the reduction of Proposition 2.

Proof (of Theorem 4). First, we show that the synthesis problem is in EXPTIME. By Proposition 2, we can construct a game (G', Ω') with $\Omega' = \cap_{m=1}^n \Omega'_m$, such that for all m , $\Omega'_m \in \{\text{Buchi, CoBuchi}\}$, and the size of the game is exponential with $O(|V| \cdot (\lambda^2 \cdot W)^n)$ vertices and $O(|E| \cdot (\lambda^2 \cdot W)^n)$ edges. Recall that the intersection of co-Büchi objectives is a co-Büchi objective. It follows that G' is a generalized Büchi \cap co-Büchi game. Then, by Theorem 2 (last item), we can compute the winning sets of both players in G in $O(n^2 \cdot |V'| \cdot |E'|) = O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$ time. Moreover, since polynomial memory strategies are sufficient in G' , exponential memory strategies are sufficient in G . Finally, the EXPTIME-hardness and the necessity of exponential memory follow from Theorem 3. This establishes the result. \square

⁶ The same vertex β could be used for each objective $\Omega_m = \text{WMP}$, instead of one such vertex by component.

4 Efficient fragment with one WMP objective

In the previous section, we considered games (G, Ω) with $\Omega = \cap_{m=1}^n \Omega_m$ being any intersection of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. In this section, we focus on a particular class of games in a way to achieve a lower complexity for the synthesis problem. We do not consider the case where at least two Ω_m are WMP objectives because the synthesis problem is already EXPTIME-hard in this case (by Theorem 3). We thus focus on the intersections of exactly one⁷ objective WMP and any number of objectives of one kind in $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. Notice that this number must be fixed in the case of objectives Sup to avoid PSPACE-hardness in this case (see Remark 3 below). For the considered fragment, we show that the synthesis problem is P-complete when λ is polynomial in the size of the game structure. The latter assumption is reasonable in practical applications where one expects a positive mean-payoff in any “short” window sliding along the play.

Theorem 5. *Let (G, Ω) be an n -weighted game with objective $\Omega = \Omega_1 \cap \Gamma$ for player 1 such that $\Omega_1 = \text{WMP}$ and $\Gamma = \cap_{m=2}^n \Omega_m$ such that $\forall m, \Omega_m = \text{Inf}$ (resp. $\forall m, \Omega_m = \text{LimInf}$, $\forall m, \Omega_m = \text{LimSup}$, $\{\forall m, \Omega_m = \text{Sup} \text{ and } n \text{ is fixed}\}$). Then the synthesis problem is decidable (in time polynomial in the size of the game, λ and $\lceil \log(W) \rceil$). In general, both players require finite-memory strategies, and pseudo-polynomial memory is sufficient for both players. Moreover, when λ is polynomial in the size of the game then the synthesis problem is P-complete.*

To prove this theorem, we use the polynomial reduction of Proposition 1 to obtain a game $(G', \Omega'_1 \cap \Gamma')$ (with $\Gamma' = \cap_{m=2}^n \Omega'_m$) such that $\Omega'_1 = \text{WMP}$ and $\forall m, \Omega'_m \in \{\text{Reach}, \text{Safe}, \text{Buchi}, \text{CoBuchi}\}$. Recall that the intersection of safety (resp. co-Büchi) objectives is a safety (resp. co-Büchi) objective. We thus have to study $\text{WMP} \cap \text{Safe}$ (resp. $\text{WMP} \cap \text{Reach}$, $\text{WMP} \cap \text{GenReach}$ (with n fixed), $\text{WMP} \cap \text{Buchi}$, $\text{WMP} \cap \text{GenBuchi}$, $\text{WMP} \cap \text{CoBuchi}$) games. Each of these six possible cases for Ω'_2 is studied in a separate section; the proof of Theorem 5 will follow.

Table 3 gives an overview of the properties of the six studied games. This table indicates time polynomial in the size of the game, λ and $\lceil \log(W) \rceil$, and pseudo-polynomial memories for both players. When λ is polynomial in size of the game, the complexity becomes polynomial. Moreover, having pseudo-polynomial memories is not really a problem since the proofs show that the strategies can be efficiently encoded by programs using two counters, as in the case of Theorem 3.

Objective	Algorithmic complexity	Player 1 memory	Player 2 memory
WMP \cap Safe	$O(\lambda \cdot V \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot W)$	$O(\lambda^2 \cdot W \cdot V)$
WMP \cap Reach	$O(\lambda \cdot V \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot W \cdot V)$	$O(\lambda^2 \cdot W \cdot V)$
WMP \cap GenReach (i fixed)	$O(\lambda \cdot 2^{2i} \cdot V \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot 2^i \cdot W \cdot V)$	$O(\lambda^2 \cdot 2^i \cdot W \cdot V)$
WMP \cap Buchi	$O(\lambda \cdot V ^2 \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot W \cdot V)$	$O(\lambda^2 \cdot W \cdot V ^2)$
WMP \cap GenBuchi	$O(\lambda \cdot i^3 \cdot V ^2 \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot W \cdot i \cdot V)$	$O(\lambda^2 \cdot W \cdot i^2 \cdot V ^2)$
WMP \cap CoBuchi	$O(\lambda \cdot V ^2 \cdot (V + E) \cdot \lceil \log(\lambda \cdot W) \rceil)$	$O(\lambda^2 \cdot W \cdot V ^2)$	$O(\lambda^2 \cdot W \cdot V)$

Table 3. Overview of the fragment (i is the number of objectives in the intersection of reachability/Büchi objectives)

4.1 Objective $\text{WMP}(\lambda, 0) \cap \text{Safe}(U)$

We begin with the easy case $\Omega'_2 = \text{Safe}$. To solve this game, we first compute the winning set X_1 for the objective $\text{Safe}(U)$ and then the winning set in the subgame $G[X_1]$ for the objective $\text{WMP}(\lambda, 0)$.

Proposition 4. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and assume that Ω is the objective $\text{WMP}(\lambda, 0) \cap \text{Safe}(U)$. Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and strategies with memory in $O(\lambda^2 \cdot W)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|)$) are sufficient for player 1 (resp. player 2).*

⁷ the case with no WMP objective will be treated in the next section.

Proof. In a first step we compute the winning set $X_1 = \text{Win}_1^{\text{Safe}(U)}(G)$ and $X_2 = V \setminus X_1$ in the game $(G, \text{Safe}(U))$, and in a second step we compute the winning set $Y_1 = \text{Win}_1^{\text{WMP}(\lambda, 0)}(G[X_1])$ and $Y_2 = X_1 \setminus Y_1$ in the game $(G[X_1], \text{WMP}(\lambda, 0))$. Notice that $X_1 \subseteq U$, and $G[X_1]$ is a game structure since X_1 is 2-closed by Corollary 1. Let us prove that $Y_1 = \text{Win}_1^\Omega(G)$ by showing that $Y_1 \subseteq \text{Win}_1^\Omega(G)$ and $X_2 \cup Y_2 \subseteq \text{Win}_2^\Omega(G)$.

Let $v \in Y_1$, and σ_1 be a winning strategy of player 1 for the objective $\text{WMP}(\lambda, 0)$ in $G[X_1]$. Since X_1 is 2-closed and $X_1 \subseteq U$, σ_1 is also winning in G for this objective while staying in U . This shows that $v \in \text{Win}_1^\Omega(G)$.

Clearly $X_2 \subseteq \text{Win}_2^\Omega(G)$ since player 2 can force to reach $V \setminus U$ from any initial vertex $v \in X_2$. Let $v \in Y_2$. We define a strategy σ_2 for player 2 from v as follows: while staying in X_1 , play a winning strategy for the objective $\text{WMP}(\lambda, 0)$ in $G[X_1]$, and as soon as player 1 decides to go to X_2 , shift to a winning strategy to reach $V \setminus U$. This strategy σ_2 clearly shows that $v \in \text{Win}_2^\Omega(G)$.

Let us now study the time complexity and the memory requirements. We easily get from Theorems 1 and 3 that Y_1 can be computed in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and that both players have finite-memory strategies with a memory in $O(\lambda^2 \cdot W)$ for player 1 (resp. in $O(\lambda^2 \cdot W \cdot |V|)$ for player 2). \square

4.2 Objectives $\text{ICWEnd}^\lambda(U)$ and $\text{GDEnd}^\lambda(U)$

The case of games $(G, \Omega_1 \cap \Omega_2)$ with $\Omega_1 = \text{WMP}(\lambda, 0)$ and $\Omega_2 \in \{\text{Reach}, \text{GenReach}, \text{Buchi}, \text{GenBuchi}, \text{CoBuchi}\}$ (with n fixed when Ω_2 is a generalized reachability objective) is much more involved. To solve them, we need to develop some new tools generalizing the concept of p -attractor while dealing with good windows. To this end, let us introduce the next two new objectives (see Figure 4).

Definition 2. Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, $U \subseteq V$ be a set of vertices, and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. We consider the next two sets of plays:

- $\text{ICWEnd}^\lambda(U) = \{\rho \in \text{Plays}(G) \mid \exists l \in \{1, \dots, \lambda\}, \rho_l \in U, \text{ and the window at position 0 is inductively-closed in } l\},$
- $\text{GDEnd}^\lambda(U) = \{\rho \in \text{Plays}(G) \mid \exists l \geq 0, \rho_l \in U \text{ and } \rho_{[0, l]}, \text{ has a } \lambda\text{-good decomposition}\}.$

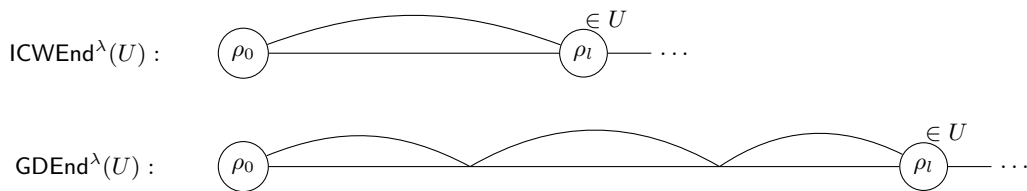


Fig. 4. A play that belongs to $\text{ICWEnd}^\lambda(U)$ and a play that belongs to $\text{GDEnd}^\lambda(U)$

Notice that the plays of $\text{ICWEnd}^\lambda(U)$ are the particular plays of $\text{GDEnd}^\lambda(U)$ such that the λ -good decomposition of $\rho_{[0, l]}$ has size 1. Hence $\text{ICWEnd}^\lambda(U) \subseteq \text{GDEnd}^\lambda(U)$. We propose two algorithms, Algorithms 1 and 2, one for computing the winning set of player 1 for the objective $\text{ICWEnd}^\lambda(U)$, and the other one for the objective $\text{GDEnd}^\lambda(U)$.

Algorithm 1 uses the next operator \oplus . Let $a, b \in \mathbb{Z} \cup \{-\infty\}$,

$$a \oplus b = \begin{cases} a + b & \text{if } a + b \geq 0 \\ -\infty & \text{otherwise} \end{cases}$$

With this definition, either $a \oplus b \geq 0$ or $a \oplus b = -\infty$. One can check that if $b \geq b'$, then $a \oplus b \geq a \oplus b'$. Moreover if $a \oplus b \geq 0$, then $a \oplus b = a + b \geq 0$. Algorithm 1 intuitively works as follows. Given a vertex v and a number i of steps, the value $C_i(v)$ is computed iteratively (from $C_{i-1}(v)$) and represents the best

total payoff that player 1 can ensure in *at most* i steps while closing the window from v in a vertex of U . The value $-\infty$ indicates that the window starting in v cannot be inductively-closed in U . Therefore, the winning set of player 1 is the set of vertices v for which $C_\lambda(v) \geq 0$. This algorithm is inspired from Algorithm GoodWin in [6] that computes $\text{Win}_1^{\text{ICWEnd}^\lambda(U)}(G)$ with $U = V$.⁸

Algorithm 1 ICWEnd

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, set $U \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\text{Win}_1^{\text{ICWEnd}^\lambda(U)}(G)$

```

1: for all  $v \in V$  do
2:   if  $v \in U$  then
3:      $C_0(v) \leftarrow 0$ 
4:   else
5:      $C_0(v) \leftarrow -\infty$ 
6:   for all  $l \in \{1, \dots, \lambda\}$  do
7:     for all  $v \in V_1$  do
8:        $C_l(v) \leftarrow \max_{(v,v') \in E} \{w(v, v') \oplus \max\{C_0(v'), C_{l-1}(v')\}\}$ 
9:     for all  $v \in V_2$  do
10:       $C_l(v) \leftarrow \min_{(v,v') \in E} \{w(v, v') \oplus \max\{C_0(v'), C_{l-1}(v')\}\}$ 
11: return  $\{v \in V \mid C_\lambda(v) \geq 0\}$ 

```

Lemma 3. *Let G be a 1-weighted game structure, U be a subset of V , and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. Then Algorithm ICWEnd computes the set $\text{Win}_1^{\text{ICWEnd}^\lambda(U)}(G)$ of winning vertices of player 1 for the objective $\text{ICWEnd}^\lambda(U)$ in $O(\lambda \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory linear in λ are sufficient for both players.*

Proof. Let us prove that Algorithm ICWEnd correctly computes the set $\text{Win}_1^{\text{ICWEnd}^\lambda(U)}(G)$. Notice that for all v and l , either $C_l(v) \geq 0$ or $C_l(v) = -\infty$.

Let v_0 be an initial vertex. Consider the following finite-memory strategy σ_1^* (resp. σ_2^*) of player 1 (resp. player 2): The memory set is $\{0, \dots, \lambda\}$, the initial memory state is λ at the initial vertex v_0 . At vertex v and memory state l , if $l \geq 1$ then player 1 (resp. player 2) chooses a vertex v' that maximizes (resp. minimizes) $w(v, v') \oplus \max\{C_0(v'), C_{l-1}(v')\}$, and if $l = 0$ then he chooses any edge. After each move, the memory state l is updated to $l - 1$ until always keeping value 0. We will show that (i) if $C_\lambda(v_0) \geq 0$ then σ_1^* is a winning strategy for $\text{ICWEnd}^\lambda(U)$ and (ii) if $C_\lambda(v_0) = -\infty$ then σ_2^* is a winning strategy for $\text{ICWEnd}^\lambda(U)$. These winning strategies are finite-memory and have both a memory in $O(\lambda)$.

(i) Take σ_1^* and suppose that $C_\lambda(v_0) \geq 0$. Let σ_2 be any strategy of player 2, and let $\rho = \rho_0 \rho_1 \dots = \text{Out}(v_0, \sigma_1^*, \sigma_2)$. To show that $\rho \in \text{ICWEnd}^\lambda(U)$, we are going to prove for all $k \in \{0, \dots, \lambda - 1\}$ that

(*) if $C_{\lambda-k}(\rho_k) \geq 0$ then there exists $l \in \{k + 1, \dots, \lambda\}$ such that $\rho_l \in U$, the window at position k is inductively-closed in l , and $\text{TP}(\rho_{[k,l]}) \geq C_{\lambda-k}(\rho_k)$.

In particular, when $k = 0$, we will get from $C_\lambda(v_0) \geq 0$ the existence of $l \in \{1, \dots, \lambda\}$ such that $\rho_l \in U$ and the window at position 0 is inductively-closed in l , that is, $\rho \in \text{ICWEnd}^\lambda(U)$.

Before proving (*), we establish the following inequality 4. Consider $C_{\lambda-k}(\rho_k) \geq 0$ with $k \in \{0, \dots, \lambda - 1\}$, and the edge (ρ_k, ρ_{k+1}) . We have

$$C_{\lambda-k}(\rho_k) \leq w(\rho_k, \rho_{k+1}) \oplus \max\{C_0(\rho_{k+1}), C_{\lambda-k-1}(\rho_{k+1})\} \quad (4)$$

Indeed if $\rho_k \in V_1$, by definition of σ_1^* , we have (4) with an equality (instead of an inequality). If $\rho_k \in V_2$, we have (4) by definition of $C_{\lambda-k}(\rho_k)$ and since player 2 does not necessarily play optimally.

⁸ We have detected a flaw in this algorithm that has been corrected in Algorithm 1. The algorithm in [6] wrongly computes the set of vertices from which player 1 can force to close the window in exactly l steps (instead of at most l steps) for some $l \in \{1, \dots, \lambda\}$.

Let us now prove (*). Suppose firstly that (4) is realized by

$$C_{\lambda-k}(\rho_k) \leq w(\rho_k, \rho_{k+1}) \oplus C_0(\rho_{k+1}). \quad (5)$$

As $C_{\lambda-k}(\rho_k) \geq 0$, it follows that $C_0(\rho_{k+1}) = 0$, that is, $\rho_{k+1} \in U$, and $w(\rho_k, \rho_{k+1}) \geq C_{\lambda-k}(\rho_k) \geq 0$. We thus get (*) with $l = k + 1$.

Suppose secondly that (4) is realized by

$$C_{\lambda-k}(\rho_k) \leq w(\rho_k, \rho_{k+1}) \oplus C_{\lambda-k-1}(\rho_{k+1}). \quad (6)$$

In this case, we prove (*) by induction on $k \in \{0, \dots, \lambda - 1\}$. We begin with $k = \lambda - 1$ (basic case of the induction). Then (6) is equal to (5) already treated. We can thus proceed with $k < \lambda - 1$. As $C_{\lambda-k}(\rho_k) \geq 0$, it follows from (6) that $C_{\lambda-k-1}(\rho_{k+1}) \geq 0$, and

$$w(\rho_k, \rho_{k+1}) + C_{\lambda-k-1}(\rho_{k+1}) \geq C_{\lambda-k}(\rho_k) \geq 0. \quad (7)$$

By induction hypothesis (with $k + 1$), from (*) and $C_{\lambda-k-1}(\rho_{k+1}) \geq 0$, we get the existence of $l \in \{k+2, \dots, \lambda\}$ such that $\rho_l \in U$, the window at position $k + 1$ is inductively-closed in l , and $\text{TP}(\rho_{[k+1,l]}) \geq C_{\lambda-k-1}(\rho_{k+1})$. To get (*) for k , with this index l , it remains to prove that $\text{TP}(\rho_{[k,l]}) \geq C_{\lambda-k}(\rho_k) \geq 0$. This follows from $\text{TP}(\rho_{[k,l]}) = w(\rho_k, \rho_{k+1}) + \text{TP}(\rho_{[k+1,l]}) \geq w(\rho_k, \rho_{k+1}) + C_{\lambda-k-1}(\rho_{k+1})$ and (7).

(ii) Take σ_2^* and suppose that $C_\lambda(v_0) = -\infty$. We have to prove that σ_2^* is a winning strategy for $\text{ICWEnd}^\lambda(U)$. Assume the contrary: then there exists a strategy σ_1 of player 1 such that the play $\rho = \text{Out}(v_0, \sigma_1, \sigma_2^*)$ belongs to $\text{ICWEnd}^\lambda(U)$. Therefore, there exists $l \in \{1, \dots, \lambda\}$ such that $\rho_l \in U$ and the window at position 0 is inductively-closed in l . To have a contradiction, we will show that $C_\lambda(v_0) \geq 0$. More generally, we are going to prove by induction on $k \in \{0, \dots, l - 1\}$ that

$$C_{\lambda-k}(\rho_k) \geq \text{TP}(\rho_{[k,l]}). \quad (8)$$

In particular, with $k = 0$, we will get $C_\lambda(v_0) \geq \text{TP}(\rho_{[0,l]}) \geq 0$.

Consider $C_{\lambda-k}(\rho_k)$ with $k \in \{0, \dots, l - 1\}$, and the edge (ρ_k, ρ_{k+1}) . We have

$$C_{\lambda-k}(\rho_k) \geq w(\rho_k, \rho_{k+1}) \oplus \max\{C_0(\rho_{k+1}), C_{\lambda-k-1}(\rho_{k+1})\}. \quad (9)$$

Indeed if $\rho_k \in V_2$, by definition of σ_2^* , we have (9) with an equality. If $\rho_k \in V_1$, we have (8) by definition of $C_{\lambda-k}(\rho_k)$ and since player 1 does not necessarily play optimally. Suppose that $k = l - 1$ (basic case of the induction). By (9) and since $\rho_l \in U$ and the window at position 0 is inductively-closed in l , we get that $C_{\lambda-l+1}(\rho_k) \geq w(\rho_{l-1}, \rho_l) \oplus C_0(\rho_l) = w(\rho_{l-1}, \rho_l)$, thus proving (8) for $k = l - 1$. Suppose now that $k < l - 1$. By (9) and induction hypothesis (with $k + 1$), we have

$$C_{\lambda-k}(\rho_k) \geq w(\rho_k, \rho_{k+1}) \oplus C_{\lambda-k-1}(\rho_{k+1}) \geq w(\rho_k, \rho_{k+1}) \oplus \text{TP}(\rho_{[k+1,l]}). \quad (10)$$

As the window at position 0 is inductively-closed in l , $\text{TP}(\rho_{[k,l]}) = w(\rho_k, \rho_{k+1}) + \text{TP}(\rho_{[k+1,l]}) \geq 0$, and then $\text{TP}(\rho_{[k,l]}) = w(\rho_k, \rho_{k+1}) \oplus \text{TP}(\rho_{[k+1,l]})$. Thus, we get with (10) that $C_{\lambda-k}(\rho_k) \geq \text{TP}(\rho_{[k,l]})$ and then (8) holds for k .

It remains to discuss the complexity of the algorithm. Clearly, it takes a number of elementary arithmetic operations which is bounded by $O(\lambda \cdot (|V| + |E|))$. As each $C_i(v)$ is bounded by $\lambda \cdot W$, each elementary arithmetic operation takes time linear in the number of bits of the encoding of $\lambda \cdot W$. Hence, the time complexity of the algorithm is in $O(\lambda \cdot (|V| + |E|) \cdot \log_2(\lambda \cdot W))$ time. \square

We now turn to Algorithm 2 for computing the winning set of player 1 for the objective $\text{GDEnd}^\lambda(U)$. It shares similarities with the algorithm computing the p -attractor $\text{Attr}_p(U, G)$ while requiring to use previous Algorithm $\text{ICWEnd}^\lambda(U)$.

Lemma 4. *Let G be a 1-weighted game structure, U be a subset of V , and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. Then Algorithm GDEnd computes the set $\text{Win}_1^{\text{GDEnd}^\lambda(U)}(G)$ of winning vertices of player 1 for the objective $\text{GDEnd}^\lambda(U)$ in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W)$) are sufficient for player 1 (resp. player 2).*

Algorithm 2 GDEnd

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, subset $U \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\text{Win}_1^{\text{GDEnd}^\lambda(U)}(G)$

```
1:  $k \leftarrow 0$ 
2:  $X_0 \leftarrow U$ 
3: repeat
4:    $X_{k+1} \leftarrow X_k \cup \text{ICWEnd}(G, X_k, \lambda)$ 
5:    $k \leftarrow k + 1$ 
6: until  $X_k = X_{k-1}$ 
7: return  $X_k$ 
```

Proof. Let $X^* = \cup_{k \geq 0} X_k$ be the set computed by Algorithm GDEnd.

Suppose that $v_0 \in X^*$, and let us prove that $v_0 \in \text{Win}_1^{\text{GDEnd}^\lambda(U)}(G)$ by induction on k such that $v_0 \in X_k$. This is trivially true for $k = 0$ since $X_0 = U$. Let $k > 0$ be such that $v_0 \in X_k \setminus X_{k-1}$. Then $v_0 \in \text{Win}_1^{\text{ICWEnd}^\lambda(X_{k-1})}(G)$ by Lemma 3. We propose the following strategy σ_1 of player 1 from v_0 : play a winning strategy for the objective $\text{ICWEnd}^\lambda(X_{k-1})$, and when this objective is realized play a winning strategy for the objective $\text{GDEnd}^\lambda(U)$ (the second strategy exists by induction hypothesis). Clearly, σ_1 is a winning strategy for $\text{GDEnd}^\lambda(U)$ showing that $v_0 \in \text{Win}_1^{\text{GDEnd}^\lambda(U)}(G)$. Notice that this winning strategy, that consists in repeatedly playing a winning strategy for the objective $\text{ICWEnd}^\lambda(X_k)$ for decreasing values of k until reaching U , is finite-memory with a memory in $O(\lambda^2 \cdot W \cdot |V|)$. Indeed, for each k (and there are at most $|V|$ such k) knowing that the objective $\text{ICWEnd}^\lambda(X_k)$ is realized requires to memorize the number of steps (up to λ) and the sum of the weights inside the current window, and to check if X_k is reached when this window becomes closed.

For the converse, we are going to prove that if $v_0 \in V \setminus X^*$ then $v_0 \in \overline{\text{Win}_2^{\text{GDEnd}^\lambda(U)}}(G)$. Let $v_0 \in V \setminus X^*$, then player 2 has a winning strategy σ_2^* for the objective $\overline{\text{ICWEnd}^\lambda(X^*)}$. Notice that against any strategy σ_1 of player 1, for $\rho = \text{Out}(v_0, \sigma_1, \sigma_2^*)$, if the window at position 0 is inductively-closed in ρ_l for $l \in \{1, \dots, \lambda\}$, then $\rho_l \in V \setminus X^*$. Consider the following strategy σ_2 of player 2 from v_0 .

1. If the current vertex v belongs to $V \setminus X^*$, play the winning strategy σ_2^* .
2. As soon as the window starting from v is first-closed in v' in l steps with $l \in \{1, \dots, \lambda\}$, as $v' \in V \setminus X^*$, go back to 1.
3. As soon as the window starting from v is bad, play whatever.

Let us show that σ_2 is a winning strategy of player 2 for the objective $\overline{\text{GDEnd}^\lambda(U)}$. Let σ_1 be any strategy of player 1 and consider $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$. Assume by contradiction that there exists $l \geq 0$ such that $\rho_l \in U$ and $\rho_{[0,l]}$ has a λ -good decomposition $(k_i)_{i=0}^l$. By Lemma 1, we can suppose that this good decomposition is maximal, that is, for each i the window at position k_i is first-closed in k_{i+1} . Therefore by construction of σ_2 , $\rho_{k_i} \in V \setminus X^*$ for all i . In particular, $\rho_l \in V \setminus X^*$ in contradiction with $\rho_l \in U$ (recall that $U \subseteq X^*$). Notice that this winning strategy is finite-memory with a memory in $O(\lambda^2 \cdot W)$. Indeed, when playing σ_2^* , knowing that the current window is first-closed or bad requires to memorize the number of steps (up to λ) and the sum of the weights inside this window.

It remains to discuss the complexity of the algorithm. The loop of Algorithm GDEnd is executed at most $|V|$ times, and each step in this loop is in $O(\lambda \cdot (|V| + |E|) \cdot \log_2(\lambda \cdot W))$ time by Lemma 3. Then, the algorithm runs in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \log_2(\lambda \cdot W))$ time. \square

Let us illustrate Algorithms 1 and 2 on the following example.

Example 2. Consider the game (G, Ω) depicted on Figure 5, where the objective $\Omega = \text{GDEnd}^2(U)$ with $U = \{v_1, v_3, v_4\}$. Let us execute Algorithm 2: $X_0 = U$, $X_1 = \text{Win}_1^{\text{ICWEnd}^2(X_0)}(G) = \{v_1, v_2, v_3, v_4\}$ and then, $X_2 = \text{Win}_1^{\text{ICWEnd}^2(X_1)}(G) = V$. Thus all vertices in G are winning for player 1 for the objective Ω . A winning strategy for player 1 consists in looping once in v_2 and then going to v_3 . Indeed for any strategy of player 2, the outcome is either $v_0 v_1^\omega$ or $v_0 v_2 v_3 v_4^\omega$, and both outcomes admit a prefix which has a 2-good decomposition and ends with a vertex of U .

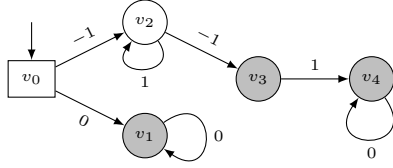


Fig. 5. Objective $\text{GDEnd}^2(U)$

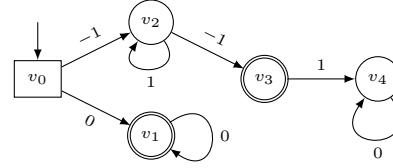


Fig. 6. Objective $\text{WMP}(2, 0) \cap \text{Reach}(U)$

4.3 Objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$

In the previous section, we have proposed an algorithm for computing the winning set of player 1 for the game $(G, \text{GDEnd}^\lambda(U))$. Thanks to this algorithm, we can compute the winning set of player 1 for the game $(G, \text{WMP}(\lambda, 0) \cap \text{Reach}(U))$ (in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$). First notice that the objectives $\text{GDEnd}^\lambda(U)$ and $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ are close to each other: a play ρ belongs to $\text{GDEnd}^\lambda(U)$ if it has a prefix which has a λ -good decomposition and ends with a vertex in U , while ρ belongs to $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ if it has a λ -good decomposition and one of its vertices belongs to U . Therefore, to solve the game $(G, \text{WMP}(\lambda, 0) \cap \text{Reach}(U))$, we first add a bit to each vertex indicating whether or not a vertex of U has been seen along the current history, and we then compute the winning set for the objective $\text{GDEnd}^\lambda(U')$ for the set U' of vertices that indicate a visit of U and are winning for the WMP objective.

Proposition 5. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and assume that Ω is the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$. Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ are sufficient for both players.*

Proof. From G and U , we construct a new game structure $G' = (V'_1, V'_2, E', w')$ with a bit added to the vertices of V that indicates whether U has been already visited (the bit equals 1) or not (the bit equals 0). More precisely, we have $V'_1 = V_1 \times \{0, 1\}$, $V'_2 = V_2 \times \{0, 1\}$; given $(v, v') \in E$, an edge $((v, b), (v', b'))$ belongs to E' such that $b' = 1$ if $(b = 1 \text{ or } v' \in U)$, and $b' = 0$ otherwise, and $w'((v, b), (v', b')) = w(v, v')$. We also define $U' = V \times \{1\}$.

For each $v \in V$, we denote by (v, b_v) the vertex of V' such that $b_v = 1$ if $v \in U$, and $b_v = 0$ otherwise. Notice that to each play ρ of G from vertex v corresponds a unique play ρ' of G' from vertex (v, b_v) . The converse is also true. Moreover, if $\rho_{[0, l]}$ is a prefix of ρ , with $l \geq 0$, such that $\rho_{[0, l]}$ has a λ -good decomposition and $\rho_k \in U$ for some $k \leq l$, then ρ' belongs to $\text{GDEnd}^\lambda(U')$. The converse is also true. Finally, a strategy on G can be easily translated to a strategy on G' and conversely. In the sequel of the proof, we will often shift from one game to the other game.

The winning set of player 1 for the objective Ω is computed as follows. In G' , we first compute the set $X' = U' \cap \text{Win}_1^{\text{WMP}(\lambda, 0)}(G')$, and then the set $Y' = \text{Win}_1^{\text{GDEnd}^\lambda(X')}(G')$. Let $Y = \{v \in V \mid (v, b_v) \in Y'\}$. We want to show that $Y = \text{Win}_1^\Omega(G)$. Notice that this set can be computed in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time by Theorem 3 and Lemma 4.

Let $v \in Y$, that is, $(v, b_v) \in Y'$. We define the next strategy σ'_1 on G' : begin with a winning strategy for the objective $\text{GDEnd}^\lambda(X')$, and as soon as this objective is realized, shift to a winning strategy for the objective $\text{WMP}(\lambda, 0)$. This strategy σ'_1 will force a play ρ' to have a λ -good decomposition $(k_i)_{i \geq 0}$ such that $\rho'_{k_i} \in U'$ for some k_i . Therefore, the strategy σ_1 on G corresponding to σ'_1 will force a play ρ to have a λ -good decomposition such that $\rho_l \in U$ for some l . Hence $v \in \text{Win}_1^\Omega(G)$. By Theorem 3 and Lemma 4 (using arguments as in the proof of this lemma), one checks that the proposed winning strategy σ_1 is finite-memory and requires a memory in $O(\lambda^2 \cdot W \cdot |V|)$.

Let $v \notin Y$, that is, $(v, b_v) \notin Y'$. We define the next strategy σ'_2 on G' : begin with a winning strategy for the objective $\text{GDEnd}^\lambda(X')$, and if the objective $\text{GDEnd}^\lambda(U')$ is realized, immediately shift to a winning strategy for the objective $\text{WMP}(\lambda, 0)$ (such a strategy exists by definition of X' and Y'). Let σ_2 be the corresponding strategy in G . Let us show that σ_2 is a winning strategy of player 2 for the objective $\overline{\Omega}$. Assume the contrary: there exists a strategy σ_1 of player 1 such that the outcome $\rho = \text{Out}(v, \sigma_1, \sigma_2)$ is

winning for Ω , that is, ρ has a λ -good decomposition $(k_i)_{i \geq 0}$, that we suppose maximal, with $\rho_l \in U$ for some l . Take the smallest index k_i such that $k_i \geq l$; then the corresponding play ρ' in G' has the same λ -good decomposition and $\rho'_{k_i} \in U'$. From this index k_i , σ'_2 will force the play to have no λ -good decomposition, in contradiction with the λ -good decomposition $(k_i)_{i \geq 0}$. Finally, as for the strategy proposed above from $v \in Y$, one checks that σ_2 is finite-memory and requires a memory in $O(\lambda^2 \cdot W \cdot |V|)$. \square

The following example illustrates the proof of Proposition 5.

Example 2 (continued). Consider the game (G, Ω) depicted on Figure 6, where $\Omega = \text{WMP}(2, 0) \cap \text{Reach}(U)$ with $U = \{v_1, v_3\}$. To compute the winning set $\text{Win}_1^\Omega(G)$, we need to modify G in a new game structure G' where we add a bit to each vertex indicating whether U has been visited (bit equals 1) or not (bit equals 0). This game structure is the one of previous Figure 5, where the set $U' = \{v_1, v_3, v_4\}$ of gray nodes are those with bit 1. We already know that every vertex of G' is winning for objective $\text{GDEnd}^2(U')$, and we note that they are also all winning for objective $\text{WMP}(2, 0)$. Therefore, in G' , player 1 can ensure a finite 2-good decomposition ending in a vertex of U' from which he can ensure an infinite 2-good decomposition. Thanks to the added bit and coming back to G , we get that all vertices of G are winning for Ω .

Objective $\text{WMP}(\lambda, 0) \cap \text{GenReach}(U_1, \dots, U_i)$

Thanks to the algorithm for the objective $\text{WMP} \cap \text{Reach}$ of Proposition 5, we here propose an algorithm for the objective $\text{WMP} \cap \text{GenReach}$ when i is fixed.

Proposition 6. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and $\Omega = \text{WMP}(\lambda, 0) \cap \text{GenReach}(U_1, \dots, U_i)$ such that i is fixed. Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot 2^{2i} \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot 2^i \cdot |V|)$ are sufficient for both players.*

The proof of this proposition uses a classical polynomial reduction (since i is fixed) of generalized reachability games to reachability games that we recall for the sake of completeness.

Lemma 5. *Each unweighted game (G, Ω) with $\Omega = \text{GenReach}(U_1, \dots, U_i)$ for a family U_1, \dots, U_i of subsets of V with i fixed can be polynomially reduced to an unweighted game (G', Ω') with $2^i \cdot |V|$ vertices and $2^i \cdot |E|$ edges, and $\Omega' = \text{Reach}(U')$ for some $U' \subseteq V'$.*

A memoryless (resp. finite-memory) strategy in G' transfers to a finite-memory strategy in G .

Proof. From G , we construct a new game G' in a way to keep in a vertex (v, b_1, \dots, b_i) of G' the current vertex v of G and i bits. For $k \in \{1, \dots, i\}$, the k^{th} bit indicates whether the set U_k has been visited (bit equals 1) or not (bit equals 0). More formally, we define $V' = V \times \{0, 1\}^i$ and we have that $((v, b_1, \dots, b_i), (v', b'_1, \dots, b'_i)) \in E'$ iff

$$(v, v') \in E \text{ and for each } k \in \{1, \dots, i\}, b'_k = \begin{cases} 1 & \text{if } b_k = 1 \text{ or } v' \in U_k \\ 0 & \text{otherwise} \end{cases}.$$

The new game G' has $2^i \cdot |V|$ vertices and $2^i \cdot |E|$ edges. We also define $U' = V \times \{1, \dots, 1\}$ and $\Omega' = \text{Reach}(U')$. Let v_0 be an initial vertex in G and b the i -tuple of bits such that the k^{th} bit is equal to 1 if $v_0 \in U_k$ and 0 otherwise. One can check that player 1 has a winning strategy from v_0 in (G, Ω) if and only if he has a winning strategy from (v_0, b) in (G', Ω') . Note that the size of G' is polynomial in the size of the original game since i is fixed.

Finally, due to the way G' is constructed from G , any strategy in G' that is memoryless (resp. finite-memory) transfers to a strategy in G that is finite-memory. \square

Proof (of Proposition 6). Let (G, Ω) be a 1-weighted game with $\Omega = \text{WMP} \cap \text{GenReach}(U_1, \dots, U_i)$. It is easy to adapt the polynomial reduction of Lemma 5 in a way to get a 1-weighted game (G', Ω') with $\Omega' = \text{WMP} \cap \text{Reach}(U')$: one just needs to define the weight function w' such that

$$w'((v, b_1, \dots, b_i), (v', b'_1, \dots, b'_i)) = w(v, v')$$

(see proof of Lemma 5). By Proposition 5, the set $\text{Win}_1^{\Omega'}(G')$ can be computed in $O(\lambda \cdot |V'| \cdot (|V'| + |E'|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V'|)$ are sufficient for both players in G' . Coming back to G , it follows that $\text{Win}_1^{\Omega}(G)$ can be computed in $O(\lambda \cdot 2^{2i} \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot 2^i \cdot |V|)$ are sufficient for both players. \square

Remark 3. The assumption about i being fixed is necessary. Indeed, by Table 2 (third row), we know that generalized reachability games are already PSPACE-hard.

4.4 Objective $\text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$

We propose in this section an algorithm for computing the winning set of player 1 for the objective $\text{WMP} \cap \text{Buchi}$. This algorithm works as follows: it repeatedly removes the set of vertices that are losing for the objective $\text{WMP} \cap \text{Reach}$, as well as its 2-attractor, until reaching a fixed point. This fixed point is the winning set for the objective $\text{WMP} \cap \text{Buchi}$.

Proposition 7. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and assume that Ω is the objective $\text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$. Then $\text{Win}_1^{\Omega}(G)$ can be computed in $O(\lambda \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|^2)$) are sufficient for player 1 (resp. player 2).*

Proof. We propose an algorithm to compute the set of winning vertices $\text{Win}_1^{\Omega}(G)$ for $\Omega = \text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$. This algorithm roughly works as follows:

1. Compute the winning set X for player 1 for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ in G .
2. Compute the 2-attractor Y of the set $V \setminus X$ in G .
3. Repeat step 1. in the game $G[V \setminus Y]$ and step 2. in the game G until X is empty or X is the set of all vertices in $G[V \setminus Y]$.
4. The final set X is the set of winning vertices.

This algorithm is detailed in Algorithm 3 (see also Figure 7). It makes several calls to two algorithms: Algorithm $\text{WMPReach}(G, U, \lambda)$ that computes the set of winning vertices of player 1 for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ and Algorithm $\text{Attractor}(G, 2, X)$ that computes the 2-attractor $\text{Attr}_2(X, G)$. To avoid heavy notation, we denote $V \setminus X$ by \overline{X} .

Algorithm 3 WMPBuchi

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, subset $U \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\text{Win}_1^{\Omega}(G)$ for $\Omega = \text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$

```

1:  $k \leftarrow 0$ 
2:  $Y_0 \leftarrow \emptyset$ 
3:  $X_0 \leftarrow \text{WMPReach}(G, U, \lambda)$ 
4: while  $X_k \neq \overline{Y}_k$  and  $X_k \neq \emptyset$  do
5:    $Y_{k+1} \leftarrow \text{Attractor}(G, 2, \overline{X}_k)$ 
6:    $X_{k+1} \leftarrow \text{WMPReach}(G[\overline{Y}_{k+1}], U, \lambda)$ 
7:    $k \leftarrow k + 1$ 
8: return  $X_k$ 

```

Let us prove that Algorithm WMPBuchi correctly computes the set $\text{Win}_1^{\Omega}(G)$. Let k^* be the first index such that $X_{k^*} = \emptyset$ or $X_{k^*} = \overline{Y}_{k^*}$. We will show that (i) if $v_0 \in X_{k^*}$ then $v_0 \in \text{Win}_1^{\Omega}(G)$, and (ii) if $v_0 \notin X_{k^*}$ then $v_0 \in \text{Win}_2^{\Omega}(G)$. We first notice that

- (*) for each vertex v in \overline{Y}_{k^*} , player 2 cannot force a play to leave \overline{Y}_{k^*} from v (Y_{k^*} is 2-closed by Theorem 1). Nevertheless player 1 could decide to go to \overline{Y}_{k^*} .

(i) Let $v_0 \in X_{k^*}$. This means that $X_{k^*} = \overline{Y}_{k^*} \neq \emptyset$. Consider the following strategy σ_1 of player 1: Play a winning strategy from v_0 for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ in the game $G[\overline{Y}_{k^*}]$. As soon as a vertex of U has been visited, and the current history has a λ -good decomposition ending in a vertex $v \in X_{k^*}$ (v belongs to X_{k^*} by (*)), play again a winning strategy from v for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$. Continue ad infinitum.

We claim that σ_1 is winning for the objective $\text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$. Indeed, consider any strategy σ_2 of player 2 and let $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$. By construction of σ_1 , ρ visits infinitely often U and has a λ -good decomposition since player 1 always plays in X_{k^*} and σ_2 cannot force to leave X_{k^*} by (*).

(ii) Let $v_0 \notin X_{k^*}$. Then according to $X_{k^*} = \emptyset$ or $X_{k^*} = \overline{Y}_{k^*}$, we have that $v_0 \in V$ or $v_0 \in Y_{k^*}$.

(a) Let us begin with case $X_{k^*} = \overline{Y}_{k^*}$. Suppose that $v_0 \in Y_k$ for some $k \in \{1, \dots, k^*\}$. We construct a strategy σ_2 from v_0 for player 2 as follows:

1. Play a memoryless strategy from v_0 to reach some $v \in \overline{X}_{k-1}$.
2. If $v \in \overline{X}_{k-1} \cap \overline{Y}_{k-1}$ and as long as the play stays in $G[\overline{Y}_{k-1}]$, play a winning strategy for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ in $G[\overline{Y}_{k-1}]$.
3. Otherwise the plays visits some $v' \in Y_{k-1}$ from which one goes back step 1. with k replaced $k-1$.

Let us show that σ_2 is winning for player 2 for the objective $\overline{\Omega}$. Let σ_1 be any strategy of player 1 and $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$. By construction of σ_2 , there exists $k' \leq k$, $k \in \{1, \dots, k^*\}$, such that ρ eventually stays in $G[\overline{Y}_{k'-1}]$, that is, there exists a smallest $i \geq 0$ such that $\rho_i \rho_{i+1} \dots \in (\overline{Y}_{k'-1})^\omega$. Moreover player 2 plays a winning strategy from ρ_i for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ which ensures that $\rho_i \rho_{i+1} \dots$ has a bad window or never visits U . It follows that ρ is losing for the objective $\text{WMP}(\lambda, 0) \cap \text{Buchi}(U)$ and that $v_0 \in \text{Win}_2^{\overline{\Omega}}(G)$.

(b) We now consider the case $X_{k^*} = \emptyset$. If $v_0 \in Y_{k^*}$, then we proceed as above. Otherwise, let $v_0 \in \overline{Y}_{k^*}$. Recall that player 2 is winning from each vertex of \overline{Y}_{k^*} for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ in $G[\overline{Y}_{k^*}]$. Therefore we define a strategy σ_2 for player 2 as above, except that we start at step 2. with $k-1 = k^*$. This strategy is thus also winning and $v_0 \in \text{Win}_2^{\overline{\Omega}}(G)$.

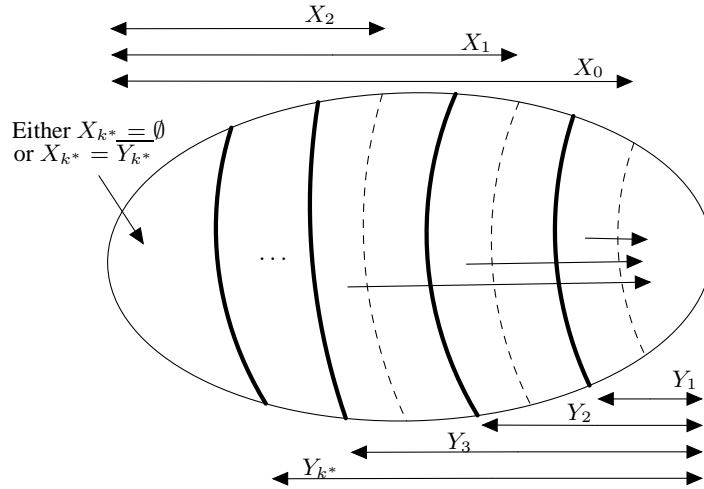


Fig. 7. Illustration of Algorithm 3

It remains to discuss the complexity of the algorithm and the memory of the winning strategies. The while loop of Algorithm WMPBuchi is executed at most $|V|$ times, and each step in this loop is in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time by Theorem 1 and Proposition 5. Then, the algorithm runs in $O(\lambda \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time. As explained in the proof, a winning strategy of player 1 consists in repeatedly playing a winning strategy for the objective $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$. This strategy is thus finite-memory and one checks that it needs a memory in $O(\lambda^2 \cdot W \cdot |V|)$ (see Proposition 5). Finally,

a winning strategy of player 2 consists in repeatedly (for decreasing values of k) playing a memoryless strategy in G and then a winning strategy for $\text{WMP}(\lambda, 0) \cap \text{Reach}(U)$ in the subgame $G[\bar{Y}_k]$. The latter strategy requires a memory in $O(\lambda^2 \cdot W \cdot |V|)$ by Proposition 5. Therefore the overall required memory is linear in $O(\lambda^2 \cdot W \cdot |V|^2)$. \square

4.5 Objective $\text{WMP}(\lambda, 0) \cap \text{GenBuchi}(U_1, \dots, U_i)$

Thanks to the algorithm for the objective $\text{WMP} \cap \text{Buchi}$ given in the previous section, we here propose an algorithm for $\text{WMP} \cap \text{GenBuchi}$.

Proposition 8. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and assume that Ω is the objective $\text{WMP}(\lambda, 0) \cap \text{GenBuchi}(U_1, \dots, U_i)$. Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot i^3 \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot i \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W \cdot i^2 \cdot |V|^2)$) are sufficient for player 1 (resp. player 2).*

The proof of this proposition uses a classical polynomial reduction of generalized Büchi games to Büchi games that we recall for the sake of completeness.

Lemma 6. *Each unweighted game (G, Ω) with $\Omega = \text{GenBuchi}(U_1, \dots, U_i)$ for a family U_1, \dots, U_i of subsets of V can be polynomially reduced to an unweighted game (G', Ω') with $i \cdot |V|$ vertices and $i \cdot |E|$ edges, and $\Omega' = \text{Buchi}(U')$ for some $U' \subseteq V'$.*

A memoryless (resp. (polynomial, exponential) memory) strategy in G' transfers to a linear memory (resp. (polynomial, exponential) memory) strategy in G .

Proof. Let us notice that if player 1 wins for the objective $\text{GenBuchi}(U_1, \dots, U_i)$, as he visits every U_k infinitely often, he visits in particular each U_k in the order $1, \dots, i$ infinitely often. Therefore from G , we construct a new game G' in a way to keep in a vertex (v, k) of G' the current vertex v of G and an integer k indicating that we have visited U_1, \dots, U_k . This integer is updated to $k + 1$ if we visit U_{k+1} , and in the special case of $k = i$, it is updated to 0. More formally, we define $V' = V \times \{0, \dots, i\}$ and

$$((v, k), (v', k')) \in E' \text{ if } (v, v') \in E \text{ and } k' = \begin{cases} 0 & \text{if } k = i \\ k + 1 & \text{if } k < i \text{ and } v' \in U_{k+1} \\ k & \text{otherwise} \end{cases}$$

We also define $U' = V \times \{0\}$ and $\Omega' = \text{Buchi}(U')$. One can check that player 1 has a winning strategy from v_0 in (G, Ω) if and only if he has a winning strategy from $(v_0, 0)$ in (G', Ω') .

Finally, due to the way G' is constructed from G , any strategy in G' that is memoryless (resp. (polynomial, exponential) memory) transfers to a strategy in G that is linear memory (resp. (polynomial, exponential) memory). \square

Proof (of Proposition 8). Let (G, Ω) be a 1-weighted game with $\Omega = \text{WMP} \cap \text{GenBuchi}(U_1, \dots, U_i)$. It is easy to adapt the polynomial reduction of Lemma 6 in a way to get a 1-weighted game (G', Ω') with $\Omega' = \text{WMP} \cap \text{Buchi}(U')$ (one just needs to define the weight function w' such that $w'((v, k), (v', k')) = w(v, v')$, see proof of Lemma 6). By Proposition 7, the set $\text{Win}_1^{\Omega'}(G')$ can be computed in $O(\lambda \cdot |V'|^2 \cdot (|V'| + |E'|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V'|)$ (resp. in $O(\lambda^2 \cdot W \cdot |V'|^2)$) are sufficient for player 1 (resp. player 2) in G' . Coming back to G , it follows that $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot i^3 \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot i \cdot |V|)$ (resp. in $O(\lambda^2 \cdot W \cdot i^2 \cdot |V|^2)$) are sufficient for player 1 (resp. player 2). \square

4.6 Objective $\text{WMP}(\lambda, 0) \cap \text{CoBuchi}(U)$

In this section, we study the last class of $\text{WMP} \cap \text{CoBuchi}$ games, and we provide an algorithm for computing the winning set of player 1 for such games. The study is elaborated; we begin with an example where we give the main ideas of the algorithm.

Example 3. Consider the game (G, Ω) depicted on Figure 8 with $\Omega = \text{WMP}(2, 0) \cap \text{CoBuchi}(U)$ and $U = \{v_0, v_2, v_4\}$. Player 1 has a winning strategy for the objective Ω if he can force the play to have a 2-good decomposition and to eventually stay in U . We first notice that any vertex winning for $\text{WMP}(2, 0) \cap \text{Safe}(U)$ is also winning for Ω , that is $\{v_4\} \subseteq \text{Win}_1^\Omega(G)$. Then, we notice that any vertex from which player 1 can force the play to

- either safely stay in U and have a 2-good decomposition,
- or have a prefix which has a 2-good decomposition and ends with a vertex that we know winning for Ω ,

is also winning for Ω . Therefore, knowing that v_4 is winning for Ω , we get that v_2 and v_3 are also winning. And then knowing that v_2, v_3, v_4 are winning, we finally get that v_1 and v_0 are winning. Hence $\text{Win}_1^{\text{WMP}(2,0) \cap \text{CoBuchi}(U)}(G) = V$.

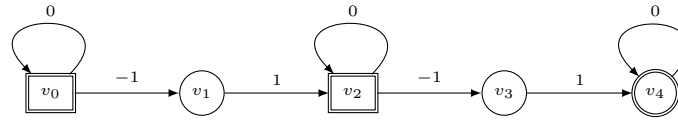


Fig. 8. A game with the objective $\Omega = \text{WMP}(2, 0) \cap \text{CoBuchi}(U)$

To formally give the algorithm and its correctness, we now need to define two specific objectives:

Definition 3. Let (V_1, V_2, E, w) be a 1-weighted game structure, $U, X, Z \subseteq V$ be three sets of vertices and $\lambda \in \mathbb{N} \setminus \{0\}$ be a window size. We consider the next sets of plays:

$$\text{End}^\lambda(U, X, Z) = \{\rho \in \text{Plays}(G) \mid \exists l \in \{1, \dots, \lambda\} \text{ such that the window at position } 0 \text{ is}$$

$$\text{inductively-closed in } l \text{ and either } (\rho_l \in \text{Win}_1^{\text{GDEnd}^\lambda(X)}(G)) \text{ or } (\rho_l \in Z \text{ and } \rho_{[0,l]} \in U^+)\},$$

$$\text{F}^\lambda(U, X) = (\text{WMP}(\lambda, 0) \cap \text{Safe}(U)) \cup \text{GDEnd}^\lambda(X).$$

Objective $\text{End}^\lambda(U, X, Z)$ asks to inductively-close a window in a vertex in $\text{GDEnd}^\lambda(X)$, or in Z while staying in U (see Figure 9). Notice that this objective is a variant of objective $\text{ICWEnd}^\lambda(U)$. The second objective $\text{F}^\lambda(U, X)$ asks for a play to be either in $\text{WMP}(\lambda, 0) \cap \text{Safe}(U)$ or in $\text{GDEnd}^\lambda(X)$.

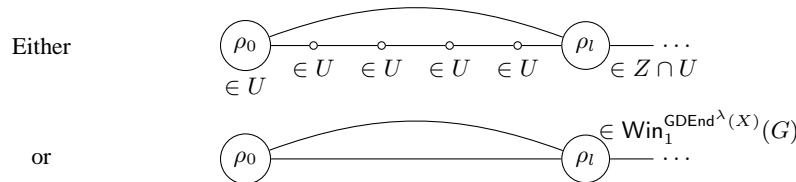


Fig. 9. A play that belongs to $\text{End}^\lambda(U, X, Z)$

The algorithm to solve $\text{WMP} \cap \text{CoBuchi}$ games roughly works as follows: compute the least fixed point (on X) of the winning set of player 1 for the objective $\text{F}^\lambda(U, X)$. We will show that this winning set is obtained by computing the greatest fixed point (on Z) of the winning set of player 1 for the objective $\text{End}^\lambda(U, X, Z)$.

We begin with a lemma showing that the winning set of player 1 for the objective $\text{End}^\lambda(U, X, Z)$ can be computed in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$. Notice that the proof of this lemma uses arguments similar to those used for Proposition 5.

Lemma 7. *Let Ω be the objective $\text{End}^\lambda(U, X, Z)$, for some subsets U, X, Z of V . Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory in $O(\lambda)$ are sufficient for both players.*

Proof. From G and U , we construct a new game structure $G' = (V'_1, V'_2, E', w')$ with a bit added to the vertices of V that indicates whether $V \setminus U$ has been already visited (the bit equals 1) or not (the bit equals 0). We denote $V \setminus U$ by \overline{U} . More precisely, we have $V'_1 = V_1 \times \{0, 1\}$, $V'_2 = V \times \{0, 1\}$; given $(v, v') \in E$, an edge $((v, b), (v', b'))$ belongs to E' such that $b' = 1$ if $(b = 1 \text{ or } v' \in \overline{U})$, and $b' = 0$ otherwise, and $w'((v, b), (v', b')) = w(v, v')$.

For each $v \in V$, we denote by (v, b_v) the vertex of V' such that $b_v = 1$ if $v \in \overline{U}$, and $b_v = 0$ otherwise.

- (*) Notice that to each play ρ of G from vertex v corresponds a unique play ρ' of G' from vertex (v, b_v) . Given a play ρ in G , if the window at position 0 is inductively-closed in l for some $l \in \{1, \dots, \lambda\}$, then it is also the case for ρ' in G' . Moreover, by construction of G' , $\rho'_l = (v', 0)$ iff $\rho_{[0, l]} \in U^+$. Finally, a strategy on G can be easily translated to a strategy on G' . For each of the above statements, the converse is also true.

The winning set of player 1 for the objective Ω is computed as follows. Let $U' = \{(v, b) \in V' \mid v \in \text{Win}_1^{\text{GDEnd}^\lambda(X)}(G) \text{ or } (b = 0 \text{ and } v \in Z)\}$. In G' , we compute the set $X' = \text{ICWEnd}^\lambda(U')$ and we let $X = \{v \in V \mid (v, b_v) \in X'\}$. Let us show that $X = \text{Win}_1^\Omega(G)$. We have that $v \in X$ iff player 1 has a winning strategy for $\text{ICWEnd}^\lambda(U')$ in G' from (v, b_v) , i.e. he can ensure that the window at position 0 in the play ρ' is inductively-closed in l with $l \in \{1, \dots, \lambda\}$ and $\rho'_l = (v', b') \in U'$. By (*) and definition of U' , it is equivalent to say that the window at position 0 in the corresponding play ρ in G is inductively-closed in l and either $\rho_l = v' \in \text{Win}_1^{\text{GDEnd}^\lambda(X)}(G)$ or $(\rho_{[0, l]} \in U^+ \text{ and } \rho_l \in Z)$. It follows that $v \in X$ iff $v \in \text{Win}_1^\Omega(G)$. Finally, the announced results about the algorithmic complexity and the memory requirements of the strategies follow from Lemma 3. \square

We now give an algorithm in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$ to compute the winning set of player 1 for the objective $F^\lambda(U, X)$ (see Algorithm 4). This algorithm uses, through a greatest fixpoint, the algorithm $\text{End}(G, \lambda, U, X, Z)$ which returns the winning set of player 1 for the objective $\text{End}^\lambda(U, X, Z)$ (see Lemma 7).

Algorithm 4 ObjF

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, subsets $U, X \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\text{Win}_1^\Omega(G)$ for $\Omega = F^\lambda(U, X)$

```

1:  $k \leftarrow 0$ 
2:  $Z_0 \leftarrow V$ 
3: repeat
4:    $T_k \leftarrow \text{End}(G, \lambda, U, X, Z_k)$ 
5:    $Z_{k+1} \leftarrow X \cup T_k$ 
6:    $k \leftarrow k + 1$ 
7: until  $Z_k = Z_{k-1}$ 
8: return  $Z_k$ 

```

Lemma 8. *Let Ω be the objective $F^\lambda(U, X)$, for some subsets U, X of V . Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time, and finite-memory strategies with memory in $O(\lambda^2 \cdot W \cdot |V|)$ are sufficient for both players.*

Proof. Let $Z^* = \bigcap_{k \geq 0} Z_k$ and $T^* = \bigcap_{k \geq 1} T_k$. We have $Z^* = X \cup T^*$ and $T^* = \text{End}^\lambda(U, X, Z^*)$. We will show that (i) if $v_0 \in Z^*$ then $v_0 \in \text{Win}_1^\Omega(G)$ and (ii) if $v_0 \notin Z^*$ then $v_0 \in \text{Win}_2^\Omega(G)$.

(i) Let $v_0 \in Z^*$. If $v_0 \in X$ then obviously v_0 is winning for the objective $\text{GDEnd}^\lambda(X)$, and thus winning for Ω . Otherwise, $v_0 \in T^*$, and we consider the next strategy σ_1 of player 1.

1. Play a winning strategy for the objective $\text{End}^\lambda(U, X, Z^*)$.
2. As soon as this objective is achieved, say in vertex v , if $v \in \text{Win}_1^{\text{GDEnd}^\lambda(X)}(G)$ then play a winning strategy for $\text{GDEnd}^\lambda(X)$, otherwise (notice that $v \in Z^*$) go back to step 1.

Let us show that σ_1 is a winning strategy of player 1 in (G, Ω) . Let $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$ with σ_2 being any strategy of player 2. By construction of σ_1 , either there exists l such that $\rho_{[0,l]}$ has a λ -good decomposition with $\rho_l \in X$, which shows that $\rho \in \text{GDEnd}^\lambda(X) \subseteq \Omega$; or we infinitely go back to step 1, in which case ρ has a λ -good decomposition and only visits vertices of U . In the latter case, it follows that $\rho \in \text{WMP}(\lambda, 0) \cap \text{Safe}(U) \subseteq \Omega$. One checks that σ_1 is a finite-memory strategy with memory in $O(\lambda^2 \cdot W \cdot |V|)$ by Lemmas 4 and 7.

(ii) We are going to prove by induction on $k \geq 1$ that if $v_0 \notin Z_k$ then $v_0 \in \text{Win}_2^{\overline{\Omega}}(G)$. Let $k = 1$ and suppose that $v_0 \notin Z_1$. Thus $v_0 \notin X$ and $v_0 \in \text{Win}_2^{\overline{\text{End}^\lambda(U, X, Z_0)}}(G)$ with $Z_0 = V$. Notice that a winning strategy of player 2 for $\overline{\text{End}^\lambda(U, X, Z_0)}$ forces a play that begin with a window first-closed in $l \in \{1, \dots, \lambda\}$, to visit $V \setminus U$ in a position $l' \leq l$ and to visit $\text{Win}_2^{\overline{\text{GDEnd}^\lambda(X)}}(G)$ at position l . We define the next strategy σ_2 of player 2: play a winning strategy for $\overline{\text{End}^\lambda(U, X, V)}$, and if the window at position 0 is first-closed in $l \in \{1, \dots, \lambda\}$, with $\rho_l \in \text{Win}_2^{\overline{\text{GDEnd}^\lambda(X)}}(G)$, switch to a winning strategy for $\text{GDEnd}^\lambda(X)$.

Let us show that σ_2 is winning for the objective $\overline{\Omega}$. Consider $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$ with σ_1 a strategy of player 1. If the window at position 0 is bad, then $\rho \notin \text{WMP}(\lambda, 0)$ and thus $\rho \in \overline{\Omega}$. Otherwise this window is first-closed in $l \in \{1, \dots, \lambda\}$. Then by construction of σ_2 , ρ visits $V \setminus U$ and from ρ_l , the play belongs to $\text{GDEnd}^\lambda(X)$. As $\rho_0 \notin X$, it follows that $\rho \notin \text{Safe}(U) \cap \text{GDEnd}^\lambda(X)$, and thus $\rho \in \overline{\Omega}$.

Suppose now that $k \geq 2$, and take $v_0 \notin Z_k$. Then, $v_0 \notin X$ and $v_0 \in \text{Win}_2^{\overline{\text{End}^\lambda(U, X, Z_{k-1})}}(G)$. Notice that a winning strategy of player 2 for $\overline{\text{End}^\lambda(U, X, Z_{k-1})}$ forces a play ρ that begin with a window first-closed in $l \in \{1, \dots, \lambda\}$, to visit $\text{Win}_2^{\overline{\text{GDEnd}^\lambda(X)}}(G)$ at position l and also to visit $V \setminus Z_{k-1}$ at position l if $\rho_{[0,l]} \in U^+$. We adapt the definition of σ_2 given in case $k = 1$ as follows: play a winning strategy for $\overline{\text{End}^\lambda(U, X, Z_{k-1})}$, and if the window at position 0 is first-closed in $l \in \{1, \dots, \lambda\}$,

- if the current history $\rho_{[0,l]}$ is in U^+ , switch to a winning strategy for $\overline{\Omega}$ (which exists by induction hypothesis as $\rho_l \notin Z_{k-1}$),
- otherwise switch to a winning strategy for $\text{GDEnd}^\lambda(X)$.

This strategy σ_2 is winning for $\overline{\Omega}$. Indeed, the arguments are the same as for case $k = 1$ except that we have to consider the additional situation where ρ has a window at position 0 first-closed in $l \in \{1, \dots, \lambda\}$ such that $\rho_{[0,l]}$ is in U^+ . Let $\rho = \rho_{[0,l-1]}\rho'$. By construction of σ_2 , we have that $\rho' \in \overline{\Omega}$, and therefore $\rho \in \overline{\Omega}$ (since $\rho_0 \notin X$).

One checks that the proposed winning strategy σ_2 is a finite-memory strategy with memory in $O(\lambda^2 \cdot W \cdot |V|)$ by Lemmas 4 and 7.

It remains to study the time complexity. Algorithm 4 runs in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time since its loop is executed at most $|V|$ times, and each step in this loop is in $O(\lambda \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time by Lemma 7. \square

We are now able to give an algorithm in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$ to solve $\text{WMP} \cap \text{CoBuchi}$ games. As expected, the algorithm (Algorithm 5) is the least fixed point (on X) of the algorithm $\text{ObjF}(G, U, X, \lambda)$.

Proposition 9. *Let $G = (V_1, V_2, E, w)$ be a 1-weighted game structure, and suppose that Ω is the objective $\text{WMP}(\lambda, 0) \cap \text{CoBuchi}(U)$ for some $U \subseteq V$ and a window size $\lambda \in \mathbb{N} \setminus \{0\}$. Then $\text{Win}_1^\Omega(G)$ can be computed in $O(\lambda \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time and strategies with memory in $O(\lambda^2 \cdot W \cdot |V|^2)$ (resp. in $O(\lambda^2 \cdot W \cdot |V|)$) are sufficient for player 1 (resp. player 2).*

Proof. Let $X^* = \cup_{k \geq 1} X_k$. We will show that (i) if $v_0 \in X^*$ then $v_0 \in \text{Win}_1^\Omega(G)$ and (ii) if $v_0 \notin X^*$ then $v_0 \in \text{Win}_2^{\overline{\Omega}}(G)$.

Algorithm 5 WMPCoBuchi

Require: 1-weighted game structure $G = (V_1, V_2, E, w)$, subset $U \subseteq V$, window size $\lambda \in \mathbb{N} \setminus \{0\}$

Ensure: $\text{Win}_1^\Omega(G)$ for $\Omega = \text{WMP}(\lambda, 0) \cap \text{CoBuchi}(U)$

```

1:  $k \leftarrow 0$ 
2:  $X_0 \leftarrow \emptyset$ 
3: repeat
4:    $X_{k+1} \leftarrow X_k \cup \text{ObjF}(G, U, X_k, \lambda)$ 
5:    $k \leftarrow k + 1$ 
6: until  $X_k = X_{k-1}$ 
7: return  $X_k$ 

```

We prove (i) by induction on k . Let $k = 1$ and suppose that $v_0 \in X_1$, then $v_0 \in \text{Win}_1^{\text{F}^\lambda(U, \emptyset)}(G)$, i.e. v_0 is winning for player 1 for the objective $\text{WMP}(\lambda, 0) \cap \text{Safe}(U)$ (see Definition 3). Therefore, a winning strategy σ_1 for this objective is clearly winning for Ω . Suppose now that $k > 1$ and $v_0 \in X_k \setminus X_{k-1}$, then $v_0 \in \text{Win}_1^{\text{F}^\lambda(U, X_{k-1})}(G)$. Consider the following strategy σ_1 of player 1: play a winning strategy from v_0 for $\text{F}^\lambda(U, X_{k-1})$ and as soon as the current history has a λ -good decomposition ending in a vertex in X_{k-1} , switch to a winning strategy for Ω (given by the induction hypothesis). This strategy σ_1 is winning for Ω . Indeed it ensures that either the play $\rho \in \text{WMP}(\lambda, 0) \cap \text{Safe}(U)$ or it decomposes as hvp' such that hv has a λ -good decomposition ending in $v \in X_{k-1}$ and vp' is winning for $\text{WMP} \cap \text{CoBuchi}$. It follows that ρ is itself winning for $\text{WMP} \cap \text{CoBuchi}$. One checks that σ_1 is a finite-memory strategy with memory in $O(\lambda^2 \cdot W \cdot |V|^2)$.

(ii) Assume that $v_0 \notin X^*$, then $v_0 \in \text{Win}_2^{\overline{\text{F}^\lambda(U, X^*)}}(G)$. by Definition 3, we have

$$\overline{\text{F}^\lambda(U, X^*)} = (\overline{\text{WMP}(\lambda, 0)} \cup \overline{\text{Safe}(U)}) \cap \overline{\text{GDEnd}^\lambda(X^*)} \quad (11)$$

Consider the following strategy σ_2 of player 2.

1. If the current vertex v belongs to $V \setminus X^*$, play a winning strategy for $\overline{\text{F}^\lambda(U, X^*)}$.
2. As soon as the current history starting from v visits $V \setminus U$ and has a λ -good decomposition, go back to step 1.

Notice that the λ -good decomposition of second step necessarily ends in a vertex of $V \setminus X^*$ by (11).

Let σ_1 be any strategy of player 1 and let $\rho = \text{Out}(v_0, \sigma_1, \sigma_2)$. We will show that $\rho \notin \Omega$. If $\rho \notin \text{WMP}(\lambda, 0)$ then $\rho \notin \Omega$. Otherwise, ρ has a λ -good decomposition $(k_i)_{i \geq 0}$ that we can suppose maximal. By (11) and definition of σ_2 , ρ has a first visit of $V \setminus U$ in some position l_1 such that for the smallest $k_{i_1} \geq l_1$ we switch from step 2. to step 1. Thus ρ has a second visit of $V \setminus U$ in l_2 such that for the smallest $k_{i_2} \geq l_2$ we switch from step 2. to step 1., ad infinitum. Therefore, ρ visits $V \setminus U$ infinitely often, that is, $\rho \notin \text{CoBuchi}(U)$, and thus $\rho \notin \Omega$. One checks that σ_2 is a finite-memory strategy with memory in $O(\lambda^2 \cdot W \cdot |V|)$.

Algorithm 5 runs in $O(\lambda \cdot |V|^2 \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time since its loop is executed at most $|V|$ times, and each step in this loop is in $O(\lambda \cdot |V| \cdot (|V| + |E|) \cdot \lceil \log_2(\lambda \cdot W) \rceil)$ time by Lemma 8. \square

Let us come back to Example 3 to illustrate the two nested fixed points.

Example 3 (continued). One can check that Algorithm WMPCoBuchi computes the following sets X_k (based on the computation of sets $Z_{k,l}$ by Algorithm ObjF):

$$\begin{array}{lll}
X_0 = \emptyset & X_1 = \{v_4\} & X_2 = \{v_2, v_3, v_4\} \\
Z_{0,0} = V & Z_{1,0} = V & Z_{2,0} = V \\
Z_{0,1} = \{v_4\} & Z_{1,1} = \{v_2, v_3, v_4\} & Z_{2,1} = V \\
Z_{0,2} = \{v_4\} & Z_{1,2} = \{v_2, v_3, v_4\} &
\end{array}$$

and finally $X_3 = X_4 = V$.

4.7 Proof of Theorem 5

We are now able to prove the main theorem of Section 4.

Proof (of Theorem 5). Let (G, Ω) be a game with objective $\Omega = \Omega_1 \cap \Gamma$ for player 1 such that $\Omega_1 = \text{WMP}$ and $\Gamma = \bigcap_{m=2}^n \Omega_m$ such that $\forall m, \Omega_m = \text{Inf}$ (resp. $\Omega_m = \text{LimInf}$, $\Omega_m = \text{LimSup}$, $\{\forall m, \Omega_m = \text{Sup}$ and n is fixed $\}$). Let us first provide an algorithm in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$ for synthesis problem. We reduce this game to a game $(G', \Omega'_1 \cap \Gamma')$ (with $\Gamma' = \bigcap_{m=2}^n \Omega'_m$) such that $\Omega'_1 = \text{WMP}$ and $\forall m, \Omega'_m \in \{\text{Reach}, \text{Safe}, \text{Buchi}, \text{CoBuchi}\}$ thanks to the polynomial reduction of Proposition 1. Recall that the intersection of safety (resp. co-Büchi) objectives is a safety (resp. co-Büchi) objective. We thus have to study six cases of games: $\text{WMP} \cap \text{Safe}$ game (resp. $\text{WMP} \cap \text{Reach}$, $\text{WMP} \cap \text{GenReach}$ (with n fixed), $\text{WMP} \cap \text{Buchi}$, $\text{WMP} \cap \text{GenBuchi}$, $\text{WMP} \cap \text{CoBuchi}$ game). The complexity and the memory requirements of each such game have been treated in Propositions 4-9 respectively (see also Table 3).

By using Proposition 1 to come back to (G, Ω) , it follows that the synthesis problem for the game (G, Ω) can be solved in time polynomial in the size of the game, λ and $\lceil \log_2(W) \rceil$ and that exponential memory strategies are sufficient for both players. More precisely, the algorithmic complexity and the memory requirements are those of Table 3 where $|V|$ is replaced by $|V| + |E|$.

The synthesis problem is P-hard because it is already P-hard for games $(G, \Omega_1) = (G, \Omega_1 \cap \Omega_2)$ with $\Omega_1 = \text{WMP}$ and $\Omega_2 = \text{Inf}$ and the weights on the second dimension all equal to 0 (see Theorem 3). Moreover, by Theorem 3, finite-memory strategies are necessary for both players. \square

5 Intersection of objectives in $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$

The aim of this section is to provide a refinement of Theorem 4 for games $(G, \Omega = \bigcap_{m=1}^n \Omega_m)$ when no objective Ω_m is a WMP objective. In this case, we get the better complexity of PSPACE-completeness (instead of EXPTIME-completeness) for the synthesis problem; nevertheless the two players still need exponential memory strategies to win (Theorem 6). We also study with precision the complexity and the memory requirements in terms of the objectives of $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ that appear in the intersection $\Omega = \bigcap_{m=1}^n \Omega_m$ (Corollary 2 below). Notice that the membership to PSPACE in Theorem 6 could have been obtained from one result proved in [1] (in this paper, the objective is defined by an LTL formula, and it is proved that deciding the winner is in PSPACE when the objective is a Boolean combination of formulas of the form “eventually p ” and “infinitely often p ”). We here propose a simple proof adapted to our context, that allows an easy study of the winning strategies as well as the identification of polynomial fragments. The proof roughly works as follows. We polynomially reduce the game G to a game G' by Proposition 1, where the objective Ω' is the intersection of a generalized-reachability, a generalized-Büchi and a co-Büchi objective. To solve this new game, we first compute in PTIME the winning set X_1 for the generalized-Büchi \cap co-Büchi objective. Then, we solve in PSPACE the generalized-reachability game in the subgame induced by X_1 (see Figure 10).

Theorem 6. *Let (G, Ω) be an n -weighted game such that $\Omega = \bigcap_{m=1}^n \Omega_m$ with $\Omega_m \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all m . Then the synthesis problem is PSPACE-complete (with an algorithm in $O(2^n \cdot (|V| + |E|))$ time) and exponential memory strategies are both necessary and sufficient for both players.*

Proof. Let us first prove that the synthesis problem is in PSPACE and that both players have exponential memory winning strategies. By Proposition 1, the game (G, Ω) is polynomially equivalent to a game (G', Ω') such that $\Omega' = \bigcap_{m=1}^n \Omega'_m$ with $\Omega'_m \in \{\text{Reach}, \text{Safe}, \text{Buchi}, \text{CoBuchi}\}$ for all m . We thus have to prove that one can decide in PSPACE whether player 1 has a winning strategy for Ω' in G' from a given vertex v_0 , and that both players have exponential memory winning strategies in G' . This proof needs several steps that are illustrated in Figure 10; to avoid heavy notation, we abusively rename (G', Ω') as (G, Ω) .

First recall that we can replace safety objectives by co-Büchi objectives and that the intersection of co-Büchi objectives is a co-Büchi objective. Moreover we can assume that each kind of objective in $\{\text{Reach}, \text{Buchi}, \text{CoBuchi}\}$ appears among the Ω_m 's (for instance the absence of a reachability objective

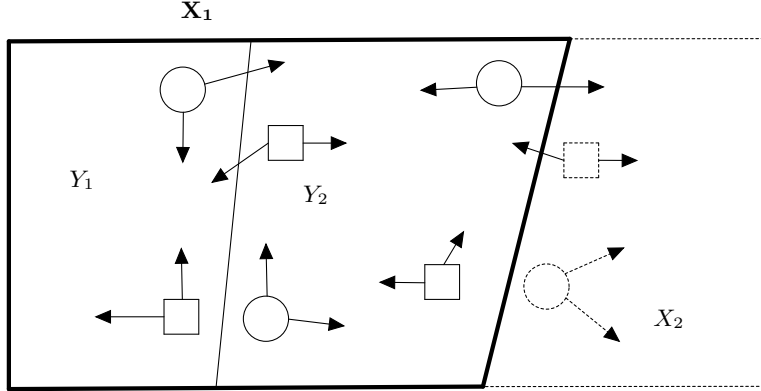


Fig. 10. Illustration of the proof

can be replaced by the presence of the objective $\text{Reach}(V)$). Therefore one can assume Ω is the intersection of one generalized reachability objective, one generalized Büchi objective and one co-Büchi objective:

$$\Omega = \text{GenReach}(U_1, \dots, U_{j-1}) \cap \text{GenBuchi}(U_j, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i) \quad (12)$$

with $1 < j < i$.

In a first step, we compute the winning set $X_1 = \text{Win}_1^{\text{GenBuchi}(U_j, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i)}(G)$ of player 1. By Theorem 2, X_1 can be computed in polynomial time, memoryless strategies suffice for player 2 whereas player 1 needs polynomial memory strategies. Moreover, by Corollary 1, X_1 is 2-closed and $X_2 = V \setminus X_1$ is 1-closed (see Figure 10). We can thus consider the subgame structure $G' = G[X_1]$.

In a second step, we compute the winning set $Y_1 = \text{Win}_1^{\text{GenReach}(U_1, \dots, U_{j-1})}(G')$ of player 1. By Theorem 2, deciding whether a vertex is in Y_1 is in PSPACE and exponential memory strategies suffice for both players. By Corollary 1, $Y_2 = X_1 \setminus Y_1$ is 1-closed in G' , and $Y_2 \cup X_2$ is 1-closed in G .

Let us show that

$$Y_1 \subseteq \text{Win}_1^{\Omega}(G). \quad (13)$$

$$X_2 \cup Y_2 \subseteq \text{Win}_2^{\overline{\Omega}}(G), \quad (14)$$

and that both players have exponential memory strategies. As X_2 , Y_2 , and Y_1 form a partition of V , the two previous inclusions are equalities, thus showing that the synthesis problem is in PSPACE (more precisely, if we recall the reduction of Proposition 1 used at the beginning of the proof, with an algorithm in $O(2^n \cdot (|V| + |E|))$ time).

We begin with inclusion (13). Let us consider the next exponential memory strategy σ_1 for player 1. Given $v \in Y_1$, player 1 first uses an exponential memory winning strategy in the game G' for the objective $\text{GenReach}(U_1, \dots, U_{j-1})$. Then as soon as all the sets U_1, \dots, U_{j-1} have been visited, he switches to a polynomial memory strategy that is winning for the objective $\text{GenBuchi}(U_j, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i)$ in the game G . Let σ_2 be any strategy of player 2 in G and let $\rho = \text{Out}(v, \sigma_1, \sigma_2)$. As X_1 is 2-closed in G , ρ cannot leave X_1 (hence σ_2 can be seen as any strategy in G'). Therefore, ρ is winning for the objective $\text{GenReach}(U_1, \dots, U_{j-1})$ in G . When each U_1, \dots, U_{j-1} has been visited by ρ , as it cannot leave X_1 and by definition of σ_1 , ρ is also winning in G for the objective $\text{GenBuchi}(U_j, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i)$. This completes the proof of inclusion (13) with exponential memory winning strategies for player 1.

Let us turn to inclusion (14). (i) Given $v \in X_2$, player 2 has a memoryless winning strategy for the objective $\overline{\text{GenBuchi}(U_j, \dots, U_{i-1}) \cap \text{CoBuchi}(U_i)}$ in the game G . This strategy is thus winning for the objective $\overline{\Omega}$. (ii) Given $v \in Y_2$, player 2 has an exponential winning strategy for the objective $\text{GenReach}(U_1, \dots, U_{j-1})$ in G' . As $Y_2 \cup X_2$ is 1-closed in G , against any strategy of player 1 in G ,

- either the play stays in Y_2 and it is then winning for player 2 for the objective $\overline{\text{GenReach}(U_1, \dots, U_{j-1})}$ in G and thus also for the objective $\overline{\Omega}$,

- or the play goes to X_2 in which case player 2 uses a strategy like in (i), and the play is again winning for player 2.

This complete the proof of inclusion (14) with exponential memory winning strategies for player 2.

It remains to prove that the synthesis problem is PSPACE-hard and that exponential memory strategies are necessary for both players. This follows from Proposition 3 and the fact that deciding the winner in generalized reachability games is PSPACE-complete and exponential memory strategies are necessary for both players (Theorem 2). \square

In the next corollary, we present several refinements (on the number of occurrences of each kind of measure) of Theorem 6. When there is at most one Sup, we get a polynomial fragment and in certain cases, players can play memoryless.

Corollary 2. *Let (G, Ω) be an n -weighted game such that $\Omega = \cap_{m=1}^n \Omega_m$ with $\Omega_m \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all m .*

- *If there is at most one m such that $\Omega_m = \text{Sup}$, then the synthesis problem is P-complete (with an algorithm in $O(n^2 \cdot (|V| + |E|) \cdot |E|)$ time), player 1 can play with polynomial memory strategies and player 2 can play memoryless.*
- *If there is no m such that $\Omega_m = \text{Sup}$ and there is exactly at most one m such that $\Omega_m = \text{LimSup}$, then the synthesis problem is P-complete (with an algorithm in $O((|V| + |E|) \cdot |E|)$ time) and player 1 can play memoryless.*
- *If there is exactly one m such that $\Omega_m = \text{Sup}$ and there is no m such that $\Omega_m \in \{\text{LimInf}, \text{LimSup}\}$, then the synthesis problem is P-complete (with an algorithm in $O(|V| + |E|)$ time) and player 1 can play memoryless.*

The proof of this corollary uses refinements of the proof of Theorem 6, except for the last item that needs a separate proof.

Proof. We begin with the proof of the first two items. As in the proof of Theorem 6, we replace safety objectives by co-Büchi objectives and the intersection of co-Büchi objectives by one co-Büchi objective. As there is at most one m such that $\Omega_m = \text{Sup}$, we get an intersection like in (12) where $\text{GenReach}(U_1, \dots, U_{j-1})$ is replaced by $\text{Reach}(U_1)$ or it disappears. With the previous proof of Theorem 6 and by Theorem 2, we can now compute $\text{Win}_1^\Omega(G) = Y_1$ in polynomial time with an algorithm in $O(n^2 \cdot (|V| + |E|) \cdot |E|)$ time. P-hardness follows from Theorem 2 and reduction of Proposition 3.

Let us study the strategies of both players.

- Suppose that there is at most one m such that $\Omega_m = \text{Sup}$. In the previous proof of inclusion (14), when $v \in Y_2$, player 2 can now play memoryless by Theorem 2.
- Suppose that there is no m such that $\Omega_m = \text{Sup}$ and there is exactly at most one m such that $\Omega_m = \text{LimSup}$. Then (12) is replaced by $\Omega = \text{Buchi}(U_1) \cap \text{CoBuchi}(U_2)$. With the proof of Theorem 6 and by Theorem 2, we see that we have an algorithm in $O((|V| + |E|) \cdot |E|)$ time and that player 1 can now play memoryless.

We now turn to the third item of the corollary: suppose that there is exactly one m such that $\Omega_m = \text{Sup}$ and there is no m such that $\Omega_m \in \{\text{LimInf}, \text{LimSup}\}$. In this case, we cannot replace safety objectives by co-Büchi objectives as in the first part of the proof. Nevertheless, we can replace the intersection of safety objectives by one safety objective, and the objective Ω is now equal to $\text{Reach}(U_1) \cap \text{Safe}(U_2)$. In a first step, we compute the winning set $X_1 = \text{Win}_1^{\text{Safe}(U_2)}(G)$ of player 1 for the objective $\text{Safe}(U_2)$, and let $X_2 = V \setminus X_1$. As X_1 is 2-closed by Corollary 1, we can consider the subgame structure $G' = G[X_1]$. In a second step, we compute the winning set $Y_1 = \text{Win}_1^{\text{Reach}(U_1)}(G')$ of player 1, and let $Y_2 = X_1 \setminus Y_1$. Let us now explain that with arguments similar to the ones used to prove Theorem 6, we have that $Y_1 = \text{Win}_1^\Omega(G)$ and that memoryless strategies are sufficient for player 1. Indeed, since X_1 is 2-closed and as $X_1 \subseteq U_2$, any memoryless winning strategy of player 1 for the objective $\text{Reach}(U_1)$ in G' is also winning for Ω in G . This shows that $Y_1 \subseteq \text{Win}_1^\Omega(G)$. It remains to show that $X_2 \cup Y_2 \subseteq \text{Win}_2^\Omega(G)$. If $v \in X_2$, player 2 plays a memoryless winning strategy for the objective $\overline{\text{Safe}}(U_2)$ in G , he is thus winning for $\overline{\Omega}$ from v . If

$v \in Y_2$, player 2 plays a memoryless winning strategy for the objective $\overline{\text{Reach}(U_1)}$ in G' , and if the play goes to X_2 , he plays a memoryless winning strategy for the objective $\text{Safe}(U_2)$ in G . We have again that player 2 is winning for Ω from v . The proposed algorithm is in $O(|V| + |E|)$ time.

Finally, P-hardness of the synthesis problem follows from Proposition 3 and Theorem 2. \square

Inf	Sup	LimInf	LimSup	Complexity class	Algorithmic complexity	Player 1 memory	Player 2 memory
any	any	any	any	PSPACE-complete	$O(2^n \cdot (V + E))$	exponential memory	exponential memory
any	≤ 1	any	any	P-complete	$O(n^2 \cdot (V + E) \cdot E)$	polynomial memory	memoryless
any	0	any	≤ 1	P-complete	$O((V + E) \cdot E)$	memoryless	memoryless
any	1	0	0	P-complete	$O(V + E)$	memoryless	memoryless

Table 4. Overview of properties for the intersection of objectives in $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$

Table 4 summarizes all the possibilities provided by Theorem 6 (first row of the table) and Corollary 2 (next rows of the table). In Example 4, we show that the results of Corollary 2 are optimal with respect to the required memory (no memory, finite memory) for the winning strategies.

Example 4. First, we come back to the game structure G depicted on Figure 1, where we only consider the second and the third dimensions. Assume $\Omega = \text{Sup}(0) \cap \text{LimSup}(0)$. Then, v_0 is winning for player 1 but memory is required to remember if player 1 has visited the edge (v_1, v_1) . The same argument holds for $\Omega = \text{Sup}(0) \cap \text{LimInf}(0)$ and $\Omega = \text{Sup}(0) \cap \text{Sup}(0)$. This example with objective $\Omega = \text{Sup}(0) \cap \text{LimSup}(0)$ indicates that player 1 cannot win memoryless in a game as in the second row of Table 4. This example with objective $\Omega = \text{Sup}(0) \cap \text{LimSup}(0)$ (resp. $\Omega = \text{Sup}(0) \cap \text{LimInf}(0)$, $\Omega = \text{Sup}(0) \cap \text{Sup}(0)$) also shows that player 1 needs memory to win if $[1, 0, 0]^9$ in the last row of Table 4 is replaced by $[1, 0, 1]$ (resp. $[1, 1, 0]$, $[2, 0, 0]$).

Now, assume that additionally $v_2 \in V_1$ in the previous game structure G (that is, G is a one-player game) and let $\Omega = \text{LimSup}(0) \cap \text{LimSup}(0)$. Again, v_0 is winning but player 1 needs memory since he has to alternate between v_1 (and take the self loop) and v_2 . This shows that in the third row of Table 4, if $[\leq 1]$ is replaced by $[2]$ then player 1 needs memory to win.

Finally, consider the game depicted on Figure 11. Let $\Omega = \text{Sup}(0) \cap \text{Sup}(0)$. Vertex v_0 is losing for player 1 (i.e. winning for player 2), but player 2 needs memory since he has to know which edge player 1 took from v_0 to counter him by taking the edge with the same vector of weights from v_3 . This shows that in the second row of Table 4, if $[\leq 1]$ is replaced by $[2]$ then player 2 needs memory to win.

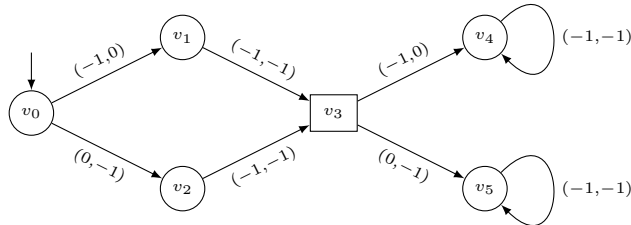


Fig. 11. Example where player 2 needs memory

⁹ This triple refers to the second, third and fourth columns of Table 4.

6 Beyond the intersection of objectives

Up to now, we have studied the synthesis problem for n -weighted games (G, Ω) such that Ω is the intersection of n objectives among $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. In this section, we go further by considering any Boolean combination of objectives instead of an intersection. The next theorem is a generalization of Theorem 4. It shows that, given a Boolean combination in a DNF or CNF form of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$, the complexity and the memory requirements do not blow up: the synthesis problem is still EXPTIME-complete and exponential memory strategies are sufficient for both players. Notice that one can assume w.l.o.g. that the dimension of the game is equal to the number of objectives that appear in the Boolean combination (by making copies of components of the weight function).

Theorem 7. *Let (G, Ω) be an nd -weighted game such that $\Omega = \bigcup_{k=1}^d \bigcap_{m=1}^n \Omega_{k,m}$ with $\Omega_{k,m} \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ for all k, m . Then, the synthesis problem is EXPTIME-complete (with an algorithm in $O(n^{d(d+2)} \cdot |V|^{d+1} \cdot |E| \cdot (\lambda^2 \cdot W)^{nd(d+2)})$ time), and exponential memory strategies are both sufficient and necessary for both players. The same result holds when $\Omega = \bigcap_{k=1}^d \bigcup_{m=1}^n \Omega_{k,m}$.*

To prove this theorem, we need to establish the next property, which is a corollary of Proposition 2.

Corollary 3. *Each n -weighted game (G, Ω) with $\Omega = \bigcap_{m=1}^n \overline{\Omega}_m$ such that for all m , $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ can be exponentially reduced to a game (G', Ω') with $O(|V| \cdot (\lambda^2 \cdot W)^n)$ vertices and $O(|E| \cdot (\lambda^2 \cdot W)^n)$ edges, and $\Omega' = \bigcap_{m=1}^n \overline{\Omega}'_m$ such that for all m , $\Omega'_m \in \{\text{Buchi}, \text{CoBuchi}\}$. A memoryless (resp. finite-memory) strategy in G' transfers to a finite-memory strategy in G .*

The proof immediately follows from the one of Proposition 2: one just needs to take the opposite of objectives Ω'_m of Proposition 2. Note that as the opposite of a Büchi (resp. co-Büchi) objective is a co-Büchi (resp. Büchi) objective, the game (G', Ω') of Corollary 3 is a generalized-Büchi \cap co-Büchi game, as in Proposition 2.

Proof (of Theorem 7). First, by Theorem 4, we know that for DNF/CNF Boolean combinations of objectives, the synthesis problem is EXPTIME-hard and that both players require exponential memory strategies. Let us show that the synthesis problem is in EXPTIME and that exponential memory strategies are sufficient for both players.

(i) Suppose first that the objective is in DNF form, i.e. $\Omega = \bigcup_{m=1}^d \bigcup_{k=1}^n \Omega_{k,m}$. Using the construction of Proposition 2, we can reduce the game (G, Ω) to an unweighted game (G', Ω') with $|V'| = O(|V| \cdot (\lambda^2 \cdot W)^{nd})$ vertices and $|E'| = O(|E| \cdot (\lambda^2 \cdot W)^{nd})$ edges, such that $\Omega' = \bigcup_{k=1}^d \bigcap_{m=1}^n \Omega'_{k,m}$ with $\Omega'_{k,m} \in \{\text{Buchi}, \text{CoBuchi}\}$. Then, Ω' is the disjunction of d generalized Büchi \cap co-Büchi objectives. Now, it is easy to adapt the polynomial reduction of generalized Büchi games to Büchi games to obtain from (G', Ω') a game (G'', Ω'') with $(n^d \cdot |V'|)$ vertices and $(n^d \cdot |E'|)$ edges, such that Ω'' is the disjunction of d Büchi \cap co-Büchi objectives (one just needs to have d counters instead of one and to define properly the new sets for the co-Büchi objectives in G'').

Therefore, we finally get a game (G'', Ω'') with $O(n^d \cdot |V| \cdot (\lambda^2 \cdot W)^{nd})$ vertices and $O(n^d \cdot |E| \cdot (\lambda^2 \cdot W)^{nd})$ edges such that Ω'' is a Rabin objective with d pairs. Let us recall that a Rabin game with d pairs can be solved in $O(x^{d+1} \cdot y \cdot dd!)$ time where x is the number of vertices and y is the number of edges, that memoryless strategies (resp. strategies with memory in $O(d!)$) are sufficient for player 1 (resp. player 2) [20]. Thus, G'' can be solved in $O(n^{d(d+2)} \cdot |V|^{d+1} \cdot |E| \cdot (\lambda^2 \cdot W)^{nd(d+2)})$ time. Finally, one can deduce from Proposition 2 that exponential memory are sufficient for both players.

(ii) Now, suppose that the objective is in CNF form, i.e. $\Omega = \bigcap_{m=1}^d \bigcup_{k=1}^n \Omega_{k,m}$. Let us take the point of view of player 2. We have $\overline{\Omega} = \bigcup_{m=1}^d \bigcap_{k=1}^n \overline{\Omega}_{k,m}$ with $\Omega_{k,m} \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. As done in (i), now using the construction of Corollary 3, we can reduce the game (G, Ω) to an unweighted game (G', Ω') such that $\overline{\Omega}'$ is a disjunction of d generalized Büchi \cap co-Büchi objectives. Using the same arguments as in (i), it suffices to solve (with the point of view of player 2) a Rabin objective with d pairs. The result follows as the game is determined. \square

Remark 4. Whereas the synthesis problem for Boolean combinations of MP and $\overline{\text{MP}}$ is undecidable [22], it is here decidable for Boolean combinations of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. Furthermore, one can show that this problem is in EXPSpace using a result of [1] as follows. With constructions

of Proposition 2 and Corollary 3, we build an exponential game (G', Ω') where Ω' is a Boolean combination of Büchi and co-Büchi objectives. Thanks to [1], games with an objective defined as a Boolean combination of formulas of the form “infinitely often p ” can be solved in PSPACE. Since the game G' is exponential, we obtain the desired result.

7 Parameterized complexity

We here provide a complete complexity analysis of our results when we fix the number n (resp. nd) of dimensions when we consider intersections (resp. DNF/CNF Boolean combinations) of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$.

Theorem 8. *Let (G, Ω) be a multi-weighted game such that*

- (a) $\Omega = \bigcap_{m=1}^n \Omega_m$, with $\Omega_m \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$,
- (b) $\Omega = \bigcap_{m=1}^n \Omega_m$, with $\Omega_m \in \{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$,
- (c) $\Omega = \bigcup_{k=1}^d \bigcap_{m=1}^n \Omega_{k,m}$, with $\Omega_{k,m} \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$.
- (d) $\Omega = \bigcap_{k=1}^d \bigcup_{m=1}^n \Omega_{k,m}$, with $\Omega_{k,m} \in \{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$.

If n is fixed (as well as nd in (c) and (d)), then the synthesis problem is

- (a, c, d) EXPTIME-complete (with a pseudo-polynomial algorithm in $O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$ time for (a) and in $O(|V|^{d+1} \cdot |E| \cdot (\lambda^2 \cdot W)^{nd(d+2)})$ time for (c, d).
- (b) P-complete (with a polynomial algorithm in $O(|V| + |E|)$ time).

Proof. For items (a), (c) and (d), the EXPTIME-hardness follows from Theorem 3. The algorithms given in Theorems 4 and 7 are now pseudo-polynomial since n (as well as nd in items (c, d)) is a fixed-parameter. Indeed, the algorithm for (a) is in $O(|V| \cdot |E| \cdot (\lambda^2 \cdot W)^{2n})$ time and the algorithm for (c, d) is in $O(|V|^{d+1} \cdot |E| \cdot (\lambda^2 \cdot W)^{nd(d+2)})$ time.

Thanks to Theorem 6, we deduce for item (b) a polynomial algorithm in $O(2^n \cdot (|V| + |E|))$ time. P-hardness follows from Theorem 2 and reduction of Proposition 3. \square

8 Conclusion

In this paper, we have studied games with Boolean combinations of heterogeneous measures in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$. The synthesis problem is EXPTIME-complete for DNF/CNF Boolean combinations and exponential memory strategies are necessary and sufficient for both players. We have focused on games with intersections of such measures and given a detailed study of the complexity and the memory requirements: EXPTIME-completeness and memory requirements still hold when the measures are taken among $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$, and we get PSPACE-completeness when WMP is not considered. In case of intersections of one WMP objective with any number of objectives of one kind among $\{\text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ (this number must be fixed in case of objectives Sup), we get P-completeness (for polynomial windows) and pseudo-polynomial strategies for both players. In case of no occurrence of WMP measure, we have proposed several refinements for which we got P-completeness and lower memory requirements.

Let us conclude with some future work. (i) Knowing that Boolean combination of mean-payoff objectives is undecidable, we have initiated in this paper the study of games with heterogeneous combination of ω -regular measures: Inf, Sup, LimInf, LimSup, and WMP. We aim at extending the study to measures that are not omega-regular, such as mean-payoff, discounted-sum, energy. (ii) The synthesis problem for arbitrary (instead of DNF/CNF) Boolean combinations of objectives in $\{\text{WMP}, \text{Inf}, \text{Sup}, \text{LimInf}, \text{LimSup}\}$ is in EXPSpace and EXPTIME-hard (see Section 3). The exact complexity should be settled. (iii) The study of Section 4 could be nicely extended with the intersection of one WMP and several other objectives of different kinds. (iv) A window version of parity objectives have been introduced in [8] and studied for unidimensional games. It could be interesting to investigate multidimensional games mixing this ω -regular objective with the objectives of this paper.

Acknowledgments. We would like to thank Mickael Randour for his availability and his valued help.

References

1. R. Alur, S. La Torre, and P. Madhusudan. Playing games with boxes and diamonds. In R. M. Amadio and D. Lugiez, editors, *CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings*, volume 2761 of *Lecture Notes in Computer Science*, pages 127–141. Springer, 2003.
2. C. Beeri. On the membership problem for functional and multivalued dependencies in relational databases. *ACM Trans. Database Syst.*, 5(3):241–259, 1980.
3. P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In F. Cassez and C. Jard, editors, *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.
4. L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J. Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.
5. K. Chatterjee, L. Doyen, T. A. Henzinger, and J. Raskin. Generalized mean-payoff and energy games. In K. Lodaya and M. Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.
6. K. Chatterjee, L. Doyen, M. Randour, and J. Raskin. Looking at mean-payoff and total-payoff through windows. *Inf. Comput.*, 242:25–52, 2015.
7. K. Chatterjee and M. Henzinger. Efficient and dynamic algorithms for alternating büchi games and maximal end-component decomposition. *J. ACM*, 61(3):15:1–15:40, 2014.
8. K. Chatterjee, T. A. Henzinger, and F. Horn. Finitary winning in omega-regular games. *ACM Trans. Comput. Log.*, 11(1), 2009.
9. K. Chatterjee, M. Randour, and J. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. *Acta Inf.*, 51(3-4):129–163, 2014.
10. L. de Alfaro and M. Faella. An accelerated algorithm for 3-color parity games with an application to timed games. In W. Damm and H. Hermanns, editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 108–120. Springer, 2007.
11. A. Degorre, L. Doyen, R. Gentilini, J. Raskin, and S. Toruńczyk. Energy and mean-payoff games with imperfect information. In A. Dawar and H. Veith, editors, *Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, August 23-27, 2010. Proceedings*, volume 6247 of *Lecture Notes in Computer Science*, pages 260–274. Springer, 2010.
12. S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games? In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 99–110. IEEE Computer Society, 1997.
13. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 368–377. IEEE Computer Society, 1991.
14. N. Fijalkow and F. Horn. Les jeux d’accessibilité généralisée. *Technique et Science Informatiques*, 32(9-10):931–949, 2013.
15. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
16. P. Hunter, G. A. Pérez, and J. Raskin. Looking at mean-payoff through foggy windows. In B. Finkbeiner, G. Pu, and L. Zhang, editors, *Automated Technology for Verification and Analysis - 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings*, volume 9364 of *Lecture Notes in Computer Science*, pages 429–445. Springer, 2015.
17. N. Immerman. Number of quantifiers is better than number of tape cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981.
18. M. Jurdzinski, R. Lazic, and S. Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 260–272. Springer, 2015.
19. D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.
20. N. Piterman and A. Pnueli. Faster solutions of rabin and streett games. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 275–284. IEEE Computer Society, 2006.

21. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 179–190. ACM Press, 1989.
22. Y. Velner. Robust multidimensional mean-payoff games are undecidable. In A. M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 312–327. Springer, 2015.
23. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, and J. Raskin. The complexity of multi-mean-payoff and multi-energy games. *Inf. Comput.*, 241:177–196, 2015.
24. Y. Velner and A. Rabinovich. Church synthesis problem for noisy input. In M. Hofmann, editor, *Foundations of Software Science and Computational Structures - 14th International Conference, FOSSACS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, volume 6604 of *Lecture Notes in Computer Science*, pages 275–289. Springer, 2011.
25. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998.
26. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.